

DISKETTE IM HEFT

64'er

über 50 Files auf Diskette

TIPS & TOOLS

Grafik

3 starke Tools für Multicolor-Bilder

Sound

Gitarrespielen im Handumdrehen

Floppy

Relative Dateien – kein Geheimnis

Programmieren

Trickkiste für Basic und Assembler

SUPER

GANZ SUPER

commodore 64 PERSONAL COMPUTER

Die Oster-Nummer!

Spitzen-Hifi-Geräte oder 3.000 Mark in bar für den besten Programmierer.

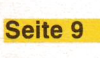


Weitere Highlights:

- »Drucker« – professionell und preiswert, der neue Panasonic.
- »3D-Construction-Kit« – wir haben das Programm getestet.
- »Scanner« – günstige PC-Scanner für den C 64 und C128.
- »Tips & Tricks« – hier zeigt sich wie's noch besser geht.
- »Tuning« – für alle, die eine tolle Kiste wollen.

Jetzt bei Ihrem Zeitschriftenhändler!

INHALT





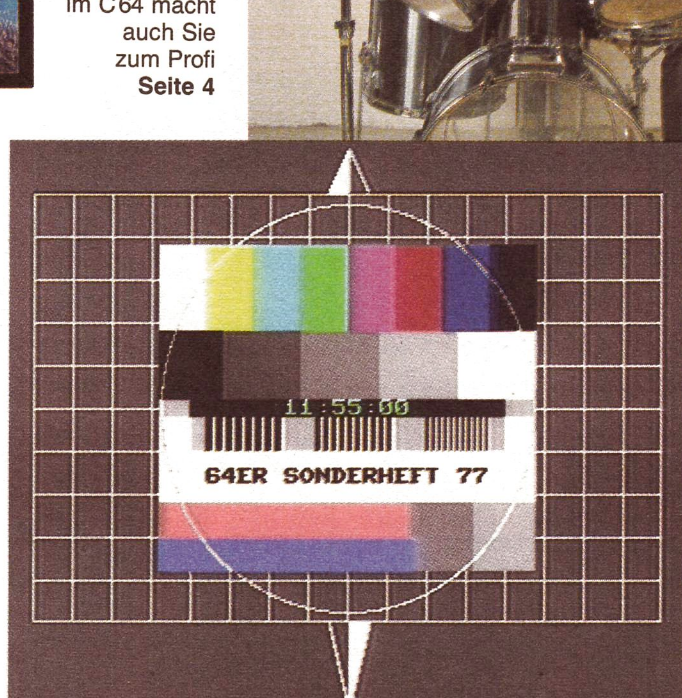
Seite 9

Seite 8

Seite 38

Seite 29

R.C.S.
Dieser Drum-Sequenzer im C 64 macht auch Sie zum Profi
Seite 4

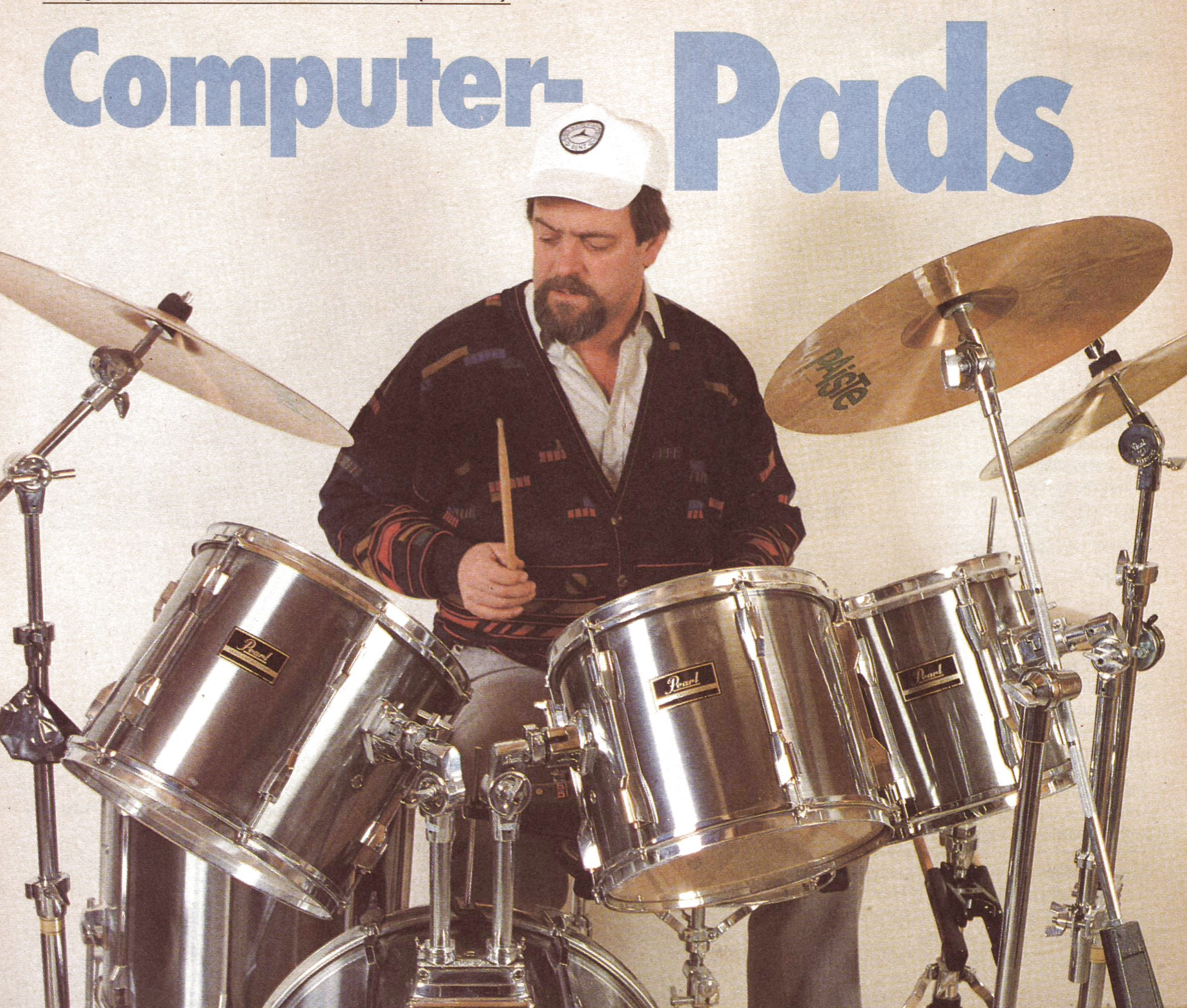


Testbild V2.1
Der optimale Abgleich von Fernseher oder Monitor wird zum Kinderspiel
Seite 49

Alle Programme zu Artikeln mit einem ▪-Symbol finden Sie auf der beiliegenden Diskette (Seite 19).

Rhythm Construction Set (R.C.S)

Computer-Pads



Wir präsentieren Ihnen ein Schlagzeug mit 17 Instrumenten und 29 vorprogrammierten Rhythmen nicht nur zum Komponieren, sondern auch zum Einüben komplizierter Schlagfolgen – ein »Muß« für jeden Drummer.

Der C 64 ist einer der universellsten Computer, dafür sorgt eine ausgeklügelte Hardware aus Videochip, Port-Bausteinen und Sound-Interface - Diesem wird allerdings viel zu wenig Beachtung geschenkt. Natürlich gibt es inzwischen bessere Sound-Generatoren, aber die hängen meist in festprogrammierten Musikcomputern oder erfordern (fast) einen Wissenschaftler für die Programmierung. Welche Möglichkeiten im VIC schlummern merkt man spätestens mit dem »R.C.S.«, dem Rhythm Construction Set. Damit steht ein komplettes Schlagzeug mit 17 Instrumenten zur Verfügung (Tabelle). Getrennt in zwei Kanäle:

1. Vordergrundinstrumente, wie Bass-Drum etc. und
2. begleitende Instrumente, wie Hi-Hat, Cabaza etc.

Pro Kanal kann zwar jeweils nur ein Instrument gespielt werden, dies wirkt sich aber im Betrieb nicht störend aus.

Im R.C.S. sind 29 Rhythmen (Pattern) vorprogrammiert (Rock, Disko, 16 Beat usw. und Schlagzeugsoli). Sie alle lassen sich beliebig verändern und und anschließend auf Diskette speichern. Jeder dieser Rhythmen besteht aus 16 Einheiten, damit sind max. 16 Schläge pro Takt erlaubt. Diese Pattern lassen sich nach dem Prinzip eines Sequenzers in einer beliebigen Reihenfolge kombinieren. Im R.C.S. sind bis zu 252 Pattern erlaubt. Dadurch lassen sich sogar extrem komplizierte Schlagfolgen komponieren.

Das Programm wird mit:

```
LOAD"R.C.S.",8
```

geladen und mit RUN gestartet. Anschließend sehen Sie das Hauptmenü (Abb. 1).

Daraus sind fünf Optionen per Tastendruck erreichbar:

<F1> – Rhythmus-Sequenzen

Mit <F1> rufen Sie den Sequenzer-Editor auf. Er fügt eine Auswahl von 29 Takten (Pattern) zu Sequenzen (Tracks) zusammen. Bis zu 252 Takte sind bei dieser Option erlaubt (Abb. 2). Das Ende-Kennzeichen («) kann wie ein Pattern übernommen werden.

<CRSR abwärts/aufwärts> – Patternwahl
... wählt eines der Pattern. Dieses wird invertiert am Bildschirm gezeigt und läßt sich mit <RETURN> in die aktuelle Taktposition übernehmen.

<CRSR rechts/links> – Taktwahl
... scrollt die aktuelle Taktposition nach rechts, bzw. links. Der Takt unter dem Pfeil (Bildschirmmitte, Abb. 2) ist gültig und seine Nummer erscheint rechts unten. In diesen Takt wird das gewählte Pattern (s. o.) übernommen.

<RETURN> – Patter in Takt
... übernimmt das gewählte Pattern in den aktuellen Takt (s. Cursortasten).

<SPACE> – Patterntest
... spielt das gewählte Pattern bis zum nächsten Tastendruck.

<INS/DEL> – Löschen eines Taktes
... löscht das gewählte Pattern an der aktuellen Taktposition.

<SHIFT INS/DEL> – Einfügen
... fügt das gewählte Pattern an der aktuellen Taktposition ein.

<CLR/HOME> – Takt Null
... ändert die aktuelle Taktposition auf »0«.

<SHIFT CLR/HOME> – löschen aller Takte
... löscht nach zweimaligem Drücken alle Takte (»00«). **Achtung:** die Tasten müssen zweimal hintereinander gedrückt werden.

<F1> – Testlauf
... spielt die Taktfolge im Speicher ab. Ein beliebiger Tastendruck schaltet wieder aus.

< + > und <-> – Geschwindigkeit
... ändert die Abspielgeschwindigkeit (nicht während des Testlaufs).

<F7> – Hauptmenü
... beendet den Sequenzen-Editor.

<F3> – Rhythmen-Muster-Editor

... erlaubt das Editieren der einzelnen Pattern. Zur Auswahl stehen 17 Instrumente für zwei Kanäle und jeweils ein Pausetakt pro Kanal (Abb. 3).

<CRSR abwärts/aufwärts> – Instrumentenwahl
... wählt eines der in der oberen Bildschirmhälfte dargestellten Instrumente. Dieses wird invertiert am Bildschirm gezeigt und läßt sich mit <RETURN> in Kanal 1 (KNL1) bzw. Kanal 2 (KNL2) übernehmen. **Achtung:** Die Instrumente »00« bis inkl. »07« werden immer in Kanal 1 übernommen, die anderen in Kanal 2.

<CRSR rechts/links> – Kanalwahl
... wählt die Taktposition im Kanal (untere Bildschirmhälfte) An dieser Positon wird das gewählte Instrument (s. o.) übernommen.

<RETURN> – Instrument in Kanal
... übernimmt das gewählte Instrument in den entsprechenden Kanal an der aktuellen Taktposition (s. Cursortasten). Die Instrumente »00« bis inkl. »07« werden immer in Kanal 1 übernommen, die anderen in Kanal 2.

<SPACE> – Instrumententest
... spielt das gewählte Instrument pro Tastendruck einmal an.

<CLR/HOME> – Taktposition Null
... ändert die aktuelle Taktposition auf den ersten Wert.

<F1> – Testlauf
... spielt das Pattern ab. Ein beliebiger Tastendruck schaltet wieder aus.

< + > und <-> – Geschwindigkeit
... ändert die Abspielgeschwindigkeit (nicht während des Testlaufs).

<,> und <.> – Geschwindigkeit
... wechselt die Nummer des aktuellen Instruments (<.> = aufsteigend, <,> = abfallend).

< = > – Kopierfunktion
... kopiert eine wählbare Patternnummer in die derzeit gültige. Hier kann dieses Pattern dann beliebig geändert werden. Die Anwahl der zu kopierenden Nummer geschieht über <,> und <.>. Mit <RETURN> wird das über Nummern gewählte Pattern in das aktuelle übernommen.

<F7> – Hauptmenü
... beendet den Rhythmen-Muster-Editor.

<F5> – Diskettenoperationen

... erlaubt Speichern oder Laden der Pattern oder Tracks und einzelne Floppyoperationen. Hier existiert ein Untermenü aus dem die einzelnen Funktionen per Tastendruck gewählt werden:

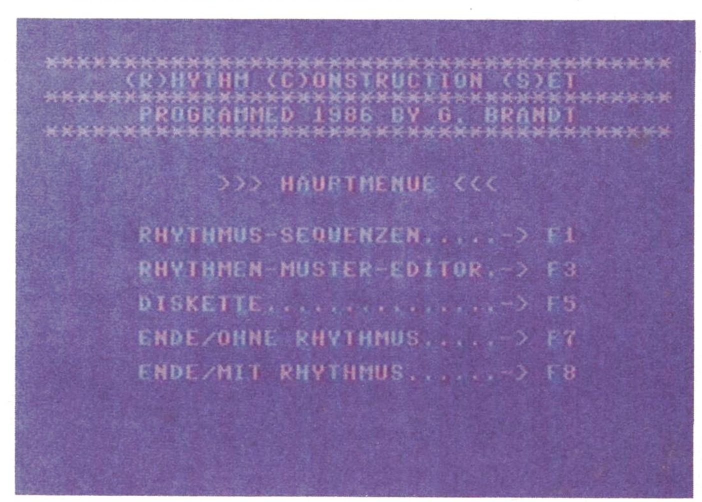
<F1> – Register speichern
... macht ein Auswahlmenü sichtbar, aus dem über den Anfangsbuchstaben die Optionen **S**equenzen, **R**hythmen, **A**lles und **I**nterrupt-File aufgerufen werden. Nach dem Aufruf einer Funktion wird der Name erfragt. Wird hier keiner angegeben und mit <RETURN> bestätigt, so unterbleibt eine Speicherung. Ansonsten wird unter dem angegebenen Namen gespeichert. Ein eventueller Diskettenfehler wird reklamiert und der Speichervorgang abgebrochen.

<F3> – Register laden
... erlaubt ein Nachladen der Interrupt-Files oder Rhythmen. Nach dem Aufruf einer Funktion wird der Name erfragt. Wird hier keiner angegeben und mit <RETURN> bestätigt, so führt das Programm keinen Ladevorgang durch. Ein eventueller Diskettenfehler wird gemeldet und der Ladevorgang abgebrochen.

<F5> – Catalog
... gibt das Inhaltsverzeichnis der Diskettenstation am Bildschirm aus.

<F6> – Disk Command/Status
... erlaubt die Eingabe von Floppykommandos. Eine leere Eingabe fragt den Fehlerkanal ab.

<F7> – Hauptmenü
... beendet das Diskettenmenü.



[1] Aus dem Hauptmenü wählen Sie per Tastendruck

<F7> – Ende ohne Rhythmus

... beendet das Programm ohne Sicherheitsabfrage. Ein Neustart ist mit RUN möglich.

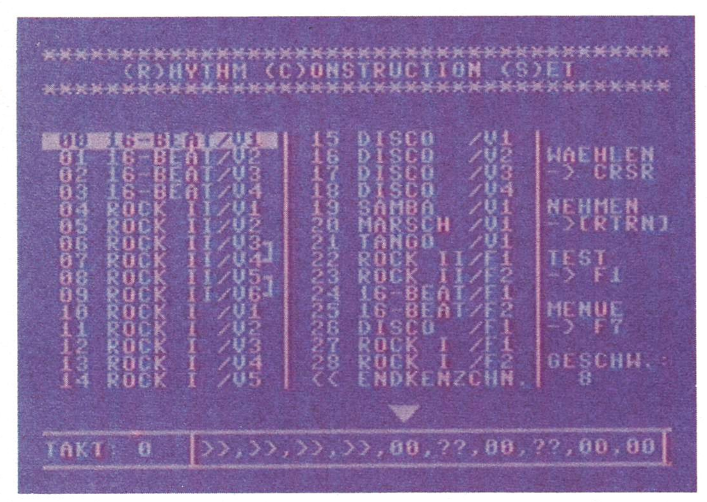
<F8> – Ende mit Rhythmus

... beendet das Programm. Das komponierte Musikstück wird dabei in den IRQ eingebunden und auch nach Verlassen des R.C.S. gespielt. Auch hier ist ein Neustart mit RUN möglich.

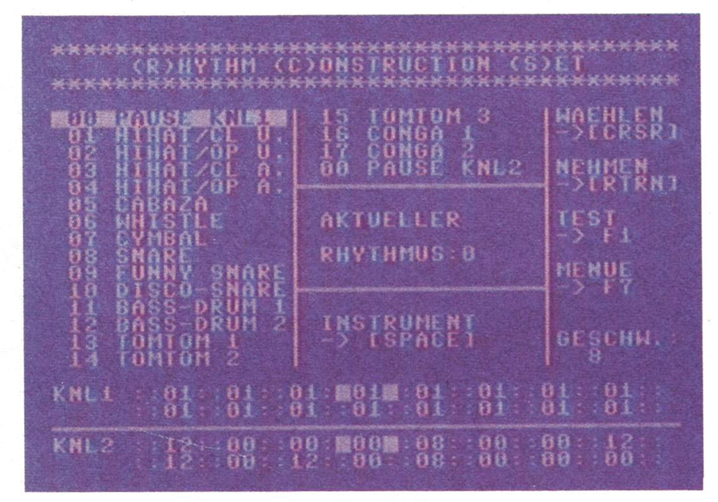
Einbau in eigene Programme

Die unter R.C.S. komponierten Rhythmen sind eigenständig lauffähig, wenn sie als Interrupt-File gespeichert wurden. Da sie in den IRQ eingebunden werden, laufen Sie unabhängig von Basicprogrammen und unabhängig von Maschinenprogrammen, die den Interrupt nicht benützen. Dadurch lassen sich diese Files in eigene Programme einbinden. Gestartet werden sie mit **SYS39242**.

Da nur zwei Stimmen (Kanäle) des SID verwendet werden, läßt sich Stimme 1 beispielsweise für den Baßlauf verwenden. Dazu muß diese allerdings mit dem Schlagzeug synchronisiert werden. R.C.S. verwaltet zu diesem Zweck einen internen Zähler, der permanent die ausgeführten Schläge mitzählt. Beim Start des IRQ-Programms wird dieser automatisch zurückgesetzt.



[2] Die Tracks werden im Sequenzen-Editor komponiert



[3] Im Muster-Editor stellen Sie die Pattern zusammen

Kanal 1		
Nr.	**Bezeichnung**	**Aufruf (SYS)**
01	Hi-Hat closed/unbetont	38255
02	Hi-Hat open/unbetont	38258
03	Hi-Hat closed/betont	38182
04	Hi-Hat open/betont	38220
05	Cabaza (Rassel)	38693
06	Whistle (Trillerpfeife)	38772
07	Cymbal (Becken)	38731
Kanal 2		
Nr.	**Bezeichnung**	**Aufruf (SYS)**
08	Snare Drum	38382
09	Funny Snare	38430
10	Disco Snare	38471
11	Bass Drum 1 (weich)	38296
12	Bass Drum (hart)	38339
13	Tom Tom 1	38517
14	Tom Tom 2	38563
15	Tom Tom 3	38609
16	Conga 1	38815
17	Conga 2	38855

Tabelle. Die Einsprungadressen der Instrumente

Das Handling eigener Bass-Sounds aus einem Basic-Programm sehen Sie aus einem Demo-Programm auf der beiliegenden Diskette. Laden Sie mit:

```
LOAD"NINETEEN",8
```

starten Sie aber noch nicht, sondern sehen Sie sich das Programm mit LIST an:

```
100 SYS64789
```

setzt die VIC-Register zurück.

```
110 A=A+1
120 IFA=1THENLOAD"NINE/I",8,8
```

In Zeile 110 wird die Variable »A« auf »1« gesetzt und demzufolge in Zeile 120 das Programm »NINE/I« unmittelbar nachgeladen, wenn A = 1. Für »NINE/I« setzen Sie später den Namen Ihres eigenen Interrupt-Files ein. Zeile 20 und die IF/THEN-Abfrage in Zeile 30 sind nötig, da der Basic-Interpreter des C64 nach dem Ladevorgang wieder zurück zum Programmbeginn springt. Ohne diese Abfrage würde in ständiger Wiederholung »NINE/I« geladen. Da aber in Zeile 110 »A« auf 2 erhöht wird trifft die IF/THEN-Aussage nicht mehr zu.

```
130 L1=54272:H1=L1+1:V1=L1+4
```

Die Speicherstellen 54272 und 54273 gehören zu den Registern des VIC. Mit beiden wird die Frequenz von Stimme 1 festgelegt. Da hier Werte zwischen 0 und 65535 erlaubt sind wurde in zwei Register (je 0 bis 255) gesplittet. In 54272 steht das Low-Byte (Wert/256) und in 54273 das High-Byte (Wert - (Low-Byte x 256)). Beide Registerpositionen werden als Variablen definiert (L1/H1). Als Variable »V1« ist Register 54276 (L1+4) bestimmt, das Kontrollregister von Stimme 1. Die beiden anderen Register sind vom R.C.S. belegt.

```
140 POKE V1+1,9:POKE V1+2,10
```

V1+1 entspricht Attack/Decay des VIC (54277). Der Wert »9«, der in dieses Register geschrieben wird, entspricht einer Anstiegszeit (Attack) von 0,002 sek. und einer Abschwellzeit (Decay) von 0,75 sek. V1+2 (54278) bestimmt Sustain/Release. Sustain bekommt hier die relative Lautstärke »0« (ist nicht benötigt, da der Ton nur angeschlagen, nicht gehalten wird). Release (Ausklingzeit) wird auf 1,5 sek. festgelegt. Mit dieser Einstellung wird ein dem Bass ähnlicher Sound für Stimme 1 bestimmt.

Die Zeilen 150 bis 170 sind lediglich für die Bildschirmgestaltung zuständig und lassen sich für Ihre Anwendungen beliebig anpassen.

```
180 SYS39242
```

startet das Interrupt-File von R.C.S. und damit den Schlagzeugsound.

In den Zeilen 190 bis 250 wird ein Zähler (Z) der am Anfang den Wert »0« hat, nach dem Einlesen von jeweils drei Daten (Data in Zeile 320 bis 350) um eins erhöht. Unter 48 wird jeweils auf die Zeile 190 zurückgeleitet (Zeile 250) und dort werden bei 24 die Datazeiger mit RESTORE auf den ersten Data-Wert zurückgesetzt. Damit werden die Data zweimal verwendet (und damit auch der Baßlauf). In Zeile 200 bekommt der Bildschirmrahmen die Farbe Rot und in Zeile 230 wieder Blau. Zwischen diesen Zeilen werden die Data in die Variablen »H«, »L« und »S« eingelesen (Zeile 210) und zwei von diesen (»H« und »L«) in die Register des VIC übertragen (Zeile 220). Mit 33 im Kontrollregister wird der Ton angeschlagen (Bit 0 = 1 ergibt den Start von Stimme 1, Bit 5 = 1 die Wellenform Sägezahn). Unmittelbar danach wird Bit 0 wieder ausgeschaltet (32) aber die Wellenform beibehalten, um den Ton ausklingen zu lassen (1,5 sek).

Zeile 240 löst eine spezielle Funktion im Interrupt-File des R.C.S. aus:

```
240 SYS39317,S
```

Es handelt sich hier um eine Warteschleife, bei der jeweils pro Taktschlag (16 pro Pattern) vom Wert »S« heruntergezählt werden. Erst wenn der Wert »0« ist, ist die Routine beendet und das Basic-Programm wird weitergeführt. Sie können damit also eine genau definierte Anzahl von Anschlägen (1 bis 255) warten und damit den Bass mit dem Schlagzeug synchronisieren.

Damit steht auch die Reihenfolge der Data ab Zeile 320 fest. Der erste von jeweils drei Werten ist das Low der Tonfrequenz, der zweite das High. Der dritte Wert bestimmt die Anzahl der Schläge (16 pro Pattern), bevor der nächste Ton angeschlagen wird.

In Zeile 260 wird der Schlagzeuglauf beendet und für Einzeltonausgabe vorbereitet:

```
SYS 39301:TI$="000000"
```

Dabei wird TI$ nur mit »000000« belegt, um später ein zeitlich definierbares Ausklingen des Tons zu ermöglichen (Zeile 290).

Zeile 270 bestimmt einen fünfmaligen Aufruf von

```
280 SYS38258
```

Damit erklingt in gleichen Abständen das Hi-Hat (Tabelle).

Wenn Sie Ihre Drum-Schule als Vorlage benutzen, läßt sich R.C.S. hervorragend zum Üben verwenden. Editieren Sie dazu mehrere Pattern mit Standard-Takten. Auf der beiliegenden Diskette befindet sich noch ein Demostück mit Drumm und Bass:

```
LOAD"AXEL F.",8
```

Mit dieser Anweisung laden Sie es. Gestartet wird mit RUN. Es lädt anschließend das File »AXEL F./I« nach. Dieses File ist eigenständig lauffähig und bringt den Drum-Sound, wenn Sie es mit

```
LOAD"AXEL F./I",8,1
```

laden und mit SYS39242 starten. Ein weiteres Interruptstück läßt sich mit

```
LOAD"NINE/I",8,1
```

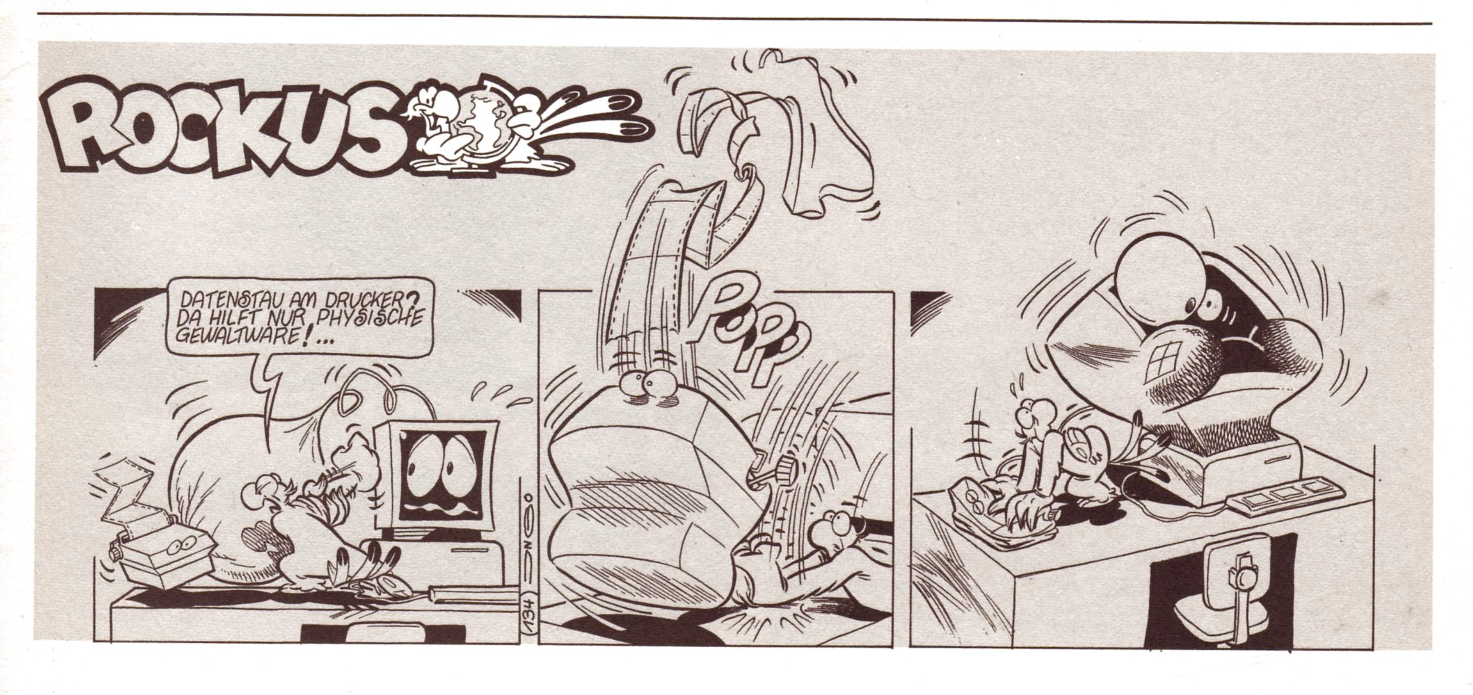
laden und ebenfalls mit SYS39242 starten. Die Files »RPS-DEMO«, »ORIGINAL-RHYTHME« und »PROGRESS-DEMO« sind Sequenzen und lassen sich daher nur im Programm verwenden. Dazu laden und starten Sie R.C.S. und wählen <F5> (Floppyoperationen) und danach <F3> Register laden. Anschließend geben Sie den Programmnamen ein. Jedes File läßt sich jetzt als Interrupt-File abspeichern. Natürlich können Sie auch die Pattern umschreiben und danach Speichern. Ihrer Phantasie sind keine Grenzen gesetzt. (gr)



[4] Das Diskmenü hat vier Optionen

Kurzinfo: Rhythm Construction Set

Programmart: Drum-Sequenzer
Laden: LOAD"R.C.S",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: erzeugt Interrupt- (11 Blocks) Sequenzer- (3 Blocks) Files, Komplett- (7 Blocks) und Rhythmus-Files (5 Blocks).
Benötigte Blocks: 25
Programmautor: Georg Brandt



21 Griffe am Bildschirm



Wir geben Ihnen ein Tool in die Hand, mit dem Sie bald zum Gipsy-King werden.

Ein lauwarmer Abend, Lagerfeuer und Gitarrenklänge – wer gerät hier nicht ins Schwärmen. Leider sitzt der Gitarrist meist bei einer anderen Fete, man nimmt statt dessen einen Kassettenrecorder und die Stimmung ist weg.

Was spricht also dagegen, sich über Zeitungsanoncen eine gebrauchte Klampfe zu besorgen (oder je nach Geldbeutel eine neue) und – selbst ist der Mann (die Frau) – das Spielen zu beginnen. Eine optimale Hilfestellung dazu bieten wir Ihnen mit »Gitarrenschule«. Geladen wird mit:

```
LOAD"GITARRENSCHULE",8
```

und gestartet mit RUN. Nach dem Nachladen eines Demomusikstücks erscheint das Hauptmenü auf dem Bildschirm (Abb. 1). Es bietet sofort die Option

Saiten stimmen

Wie in Schulungsheften üblich ist die zuoberst angezeigte Saite das hohe »E« (die dünnste Saite). Sie blicken praktisch über die Gitarre auf die Saiten. Der Ton dafür erscheint wenn Sie <SPACE> drücken. In der Reihenfolge erreichen Sie die H-Saite (hier »B«) mit <B>, die G-Saite mit <G>, »D« mit <D>, »A« mit <A> und die tiefe E-Saite mit <E>. Eine Eselsbrücke um sich die Reihenfolge der Saiten zu merken (in umgekehrter Reihenfolge) lautet:

Eine
Alte
Dame
Ging
Heim
Essen

Ein zweites Menü erreichen Sie mit <Q> (Abb. 2):

Griff üben – <Q>

... erlaubt das Üben je eines von 21 Akkorden. Zusätzlich wird dieser gespielt.

Achtung: Geben Sie einen der in den letzten Zeilen angezeigten Akkorden ein (z.B. CM) und bestätigen Sie mit <RETURN>. Verwenden Sie keine Cursortasten. Da das Programm die normale Input-Routine verwendet, erhalten Sie sonst die Anzeige: Dieser Griff ist leider nicht vorhanden.

Als Eingabe sind folgende Abkürzung gültig:

CM – C-Moll
C# – Cis
C#M – Cis-Moll
C – C
D – D
DM – D-Moll
E – E
EM – E-Moll
F – F
FM – F-Moll
F# – Fis
G – G
G7 – G-septime
GM – G-Moll
G# – Gis
A – A
A7 – A-septime
AM – A-Moll
A# – Ais
H – H

Die Griffe werden in der oberen Bildschirmhälfte gezeigt. Die Anordnung der Saiten entspricht dem Blickwinkel bei umgehängter Gitarre. Auch die Finger sind definiert. Beachten Sie die Erklärung in der Bildschirmmitte.

Um ins Hauptmenü zu gelangen geben Sie <Q> und danach <RETURN> ein.

Dort existiert eine dritte Option:

Hintergrundmusik

Sie wird (nur in diesem Menü) mit <F1> ein und mit <F7> ausgeschaltet. Achtung: Wenn Sie die Gitarre stimmen wollen, sollte die Musik ausgeschaltet sein.

Wenn Sie die »Gitarrenschule« ohne Demomusik verwenden wollen, müssen Sie einige Zeilen ändern, bzw. löschen und das Programm anschließend speichern:

Zeile 1 – löschen
Zeile 37, 38 und 39 – löschen
Zeile 150 – löschen
Zeile 160 – löschen

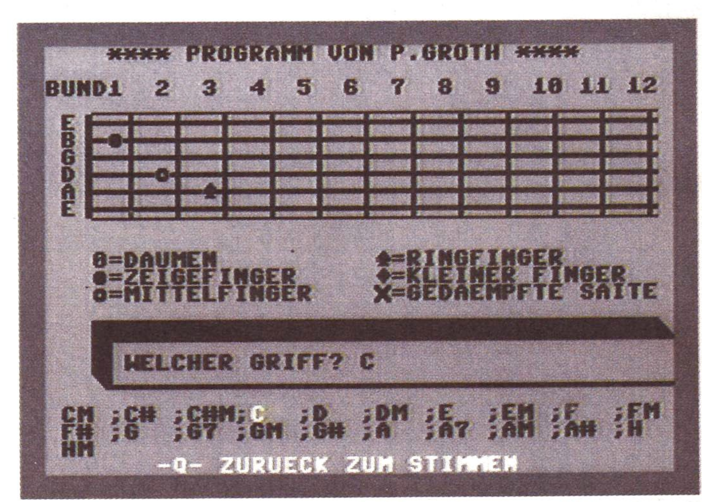
In Zeile 48 entfält »GOSUB160«, sie lautet dann:

```
48 IF A$="Q"THENGOSUB1000:GOTO200
```

Für ein schnelles Erfolgserlebnis genügen (bei einfachen Liedern) drei Griffe. Am leichtesten erlernbar sind »E«, »A« und »D«. Wenn Sie Ihre Finger für diese Griffe schnell genug umsetzen können, steht dem Begleiten von Zech- und Wanderliedern nichts mehr im Wege. (gr)



[1] Aus dem Hauptmenü läßt sich die Gitarre stimmen



[2] Im Übungsmenü sehen und hören Sie einen Akkord

Kurzinfo: Gitarrenschule

Programmart: Lernprogramm
Laden: LOAD"GITARRENSCHULE",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 30+45 (SOUND2=Demosound)
Programmautor: Peter Groth

Amica-Convert – Der Grafik-Wandler

Sechs Formate zur Wahl

Spätestens wenn Sie Ihre grafischen Werke mit den unterschiedlichen Vorzügen verschiedener Malprogramme nachbearbeiten wollen, benötigen Sie unseren Formatwechsler.

Unterschiedliche Malprogramme haben auch verschiedene Diskettenformate. Darum lassen sich Grafiken und Bilder untereinander gemeinhin nicht austauschen. Aber es funktioniert trotzdem – mit »Amica-Convert«. Dieser Multicolorkonverter wandelt Ihre Werke in sechs unterschiedliche Formate:
- Blazing Paddles
- Advanced Art Studio
- Paint Magic
- einfache Bitmap

Zusätzlich stehen Wandlungsfunktionen für
- Zeichensatz und Video-RAM, ggf. mit Farb-RAM
- und Sprites

zur Verfügung. Gerade aus diesen Funktionen ergeben sich verblüffende Möglichkeiten:

1. Sprites oder Zeichensatzgrafik kann zur Weiterverarbeitung ins Format eines gängigen Grafikprogramms verwandelt werden.

2. Eine komplette Spielegrafik läßt sich mit einem Malprogramm, (z.B. mit Amica Paint) entwerfen und in Sprites oder Zeichensatzgrafik umwandeln. Hierbei spielt die »Playfield-Funktion« eine besondere Rolle: Sie ermöglicht es, auch überbildschirmgroße Zeichensatzgrafiken, wie z.B. für eine scrollende Landschaft, zu erzeugen.

[1] Das Hauptmenü besitzt Optionen zum Laden, Nachbearbeiten und Speichern von Bildern

Geladen wird von der beiliegenden Diskette mit

`LOAD"AMICA-CONVERT",8`

Nach dem Starten (RUN) tippen Sie eine beliebige Taste an, um das Hauptmenü zu aktivieren (Abb. 1 u. 2). Hier lassen sich mit <F1> Grafiken laden, die aktuelle Grafik bearbeiten (<F3>) oder Grafik speichern bzw. drucken (<F5>). Zusätzlich bietet es mit <F7> einen Betrachtungsmodus, bei dem das ganze Bild sichtbar ist.

Im Programm führt <RUN/STOP> Sie grundsätzlich ins Hauptmenü oder zur letzten Eingabe zurück.

Bei der Eingabe von Farben werden diese entweder über die entsprechende Farbtaste (mit <CBM Zifferntaste> bzw. <Control Zifferntaste>) oder im hexadezimal Code (0-9,A-F) eingegeben. Um Ihnen einen Überblick zu verschaffen, ist am unteren Bildschirmrahmen immer ein Farbbalken mit den 15 Farben und den entsprechenden Nummern sichtbar (außer bei »<F7> - Watch«).

<**F1**> – Input (Grafik laden)

Dieses Menü (Abb. 3) enthält Ladefunktionen für die einzelnen Grafikformate. Dabei werden die einzelnen Optionen per Tastendruck ausgewählt und entsprechend auf eine Eingabe gewartet, oder andere Funktionen (z.B. »Directory«) direkt ausgeführt. Bei der Namenseingabe sind die Kennungen (Prefix, Suffix) der einzelnen Formate vorgegeben, lassen sich aber überschreiben. Dadurch wird das Einlesen bei geänderten Bildnamen ermöglicht. Beim Laden sämtlicher Dateien sind die Startadressen der Files belanglos. Jede Eingabe läßt sich mit <RUN/STOP> abbrechen. Danach schaltet das Programm zum Hauptmenü zurück.

<**A**> – Clear Screen

...löscht den aktuellen Grafikspeicher.

<**C**> – Amica-Paint

...lädt nach der Namenseingabe ein Bild im Amica-Paint-Format. Der Prefix (»[B]«) wird vorgegeben. Beim Ladevorgang wird der Grafikspeicher mit dem neuen Bild überschrieben. Das heißt: Ihr altes Gemälde wird gelöscht.

<**E**> – Advanced Art Studio

...lädt nach der Namenseingabe ein Bild im Advanced-Art-Studio-Format. Der Suffix (»MPIC«) wird dabei automatisch vorgegeben. Auch hier wird beim Ladevorgang der Grafikspeicher mit dem neuen Bild überschrieben.

<**G**> – Simple Bitmap

...lädt 8000 Bytes Grafikdaten, die als Bitmuster-Grafik im Format des VIC-Chip vorliegen müssen. Geben Sie zuerst die vier Farben für die einzelnen Bitkombinationen ein. Anschließend erwartet das Programm den Filenamen der Bitmap. Das

File sollte 32 oder 33 Blocks (entspricht 8000 bzw. 8192 Bytes) lang sein.

<**I**> - Char+Video+Color

...lädt Zeichensatz, Bildschirminhalt und Farbspeicher. Hier ist zunächst die Eingabe von drei der verwendeten Farben erforderlich. Die vierte ist in den Daten des Farbspeichers festgelegt. Es spielt keine Rolle, ob Bit 3 (zuständig für den Multicolor-Modus) im Farbspeicher gesetzt oder gelöscht ist, da es ausmaskiert wird (zuständig für »Multicolor«). Diese Option erlaubt es, Zeichensatzgrafiken zu laden und in die unterschiedlichsten Formate zu wandeln.

<**K**> - Directory

...zeigt den aktuellen Inhalt einer eingelegten Diskette an.

<**B**> - Koalapainter

...Bild im Koalapainter-Format wird nach der Namenseingabe geladen. Der Prefix »APIC« wird dabei automatisch vorgestellt und läßt sich überschreiben. Beachten Sie, daß der erste Buchstabe ein reverses »A« ist. Er läßt sich, wenn Sie ihn einmal überschrieben oder gelöscht haben (<SHIFT CLR/HOME>) nicht mehr eintippen. Beenden Sie in diesem Fall die Option mit <RUN/STOP>. Danach rufen Sie die Option neu auf (<F1>, danach <B>).

<**D**> - Paint Magic

...nach der Namenseingabe wird ein Bildformat des Malprogramms Paint Magic geladen.

<**F**> - Blazing Paddles

...lädt nach der Namenseingabe ein Bild im Blazing-Paddles-Format. Der Prefix »PI.« wird dabei automatisch vorgestellt und läßt sich überschreiben.

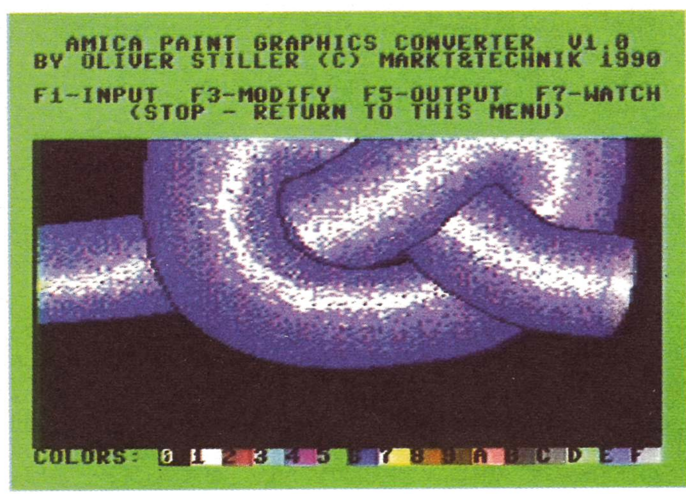
<**H**> - Char+Video-RAM

...lädt einen Bildschirminhalt und den dazugehörigen Zeichensatz. Hier ist zunächst die Eingabe der vier verwendeten Farben erforderlich. Als »Video-RAM« erwartet der Konverter eine Datei mit vier bis fünf Blöcken (1000 oder 1024 Bytes). Der Zeichensatz kann bis zu neun Blöcke (256 Zeichen) lang sein. Mit dieser Option lassen sich »Playfields« aus Spielen in Grafiken umwandeln.

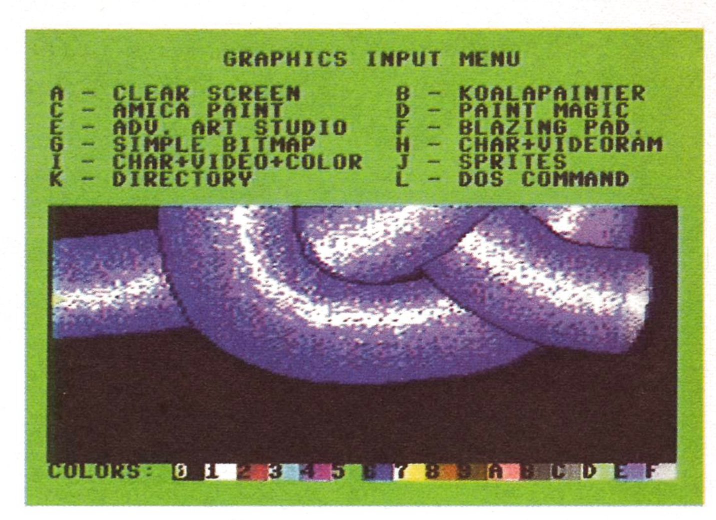
<**J**> - Sprites

...plaziert die Sprites auf der gerade sichtbaren Grafik, dabei bleiben Anteile des aktuellen Bildes erhalten. Wenn Sie Sprites nicht mit anderen Grafiken mischen wollen, sollten Sie vorher den aktuellen Speicher löschen (<A>). Für die Farben gilt:

Wenn die aktuelle Hintergrundfarbe nicht unter den eingegebenen Farben vorkommt, wird der Hintergrund neu eingefärbt, da sonst beim Plazieren der Sprites Farbprobleme (color-clashes) zu erwarten sind. Ein Plazieren der Sprites auf dem Bildschirm kann sowohl automatisch (<A>) als auch individuell (<I>) erfolgen. Im Automatikmodus werden (bis zu)



[2] Im Hauptmenü sehen Sie nur Teile des Bildes. <F7> zeigt die ganze Grafik.



[3] Im Input-Menü lassen sich die unterschiedlichsten Formate laden und darstellen

80 Sprites in einem Raster auf dem Bildschirm angeordnet. Dabei werden in horizontaler Richtung von Sprite zu Sprite jeweils ein Zeichen und in vertikaler Richtung drei Pixel freigelassen. Falls sich mehr als 80 Sprites in der Datei befinden, werden die letzten Sprites ignoriert.

Beim individuellen Plazieren sind folgende Tasten belegt:

<I>,<J>,<K>,<M> - bewegen des Cursors in Pixel-Schritten

<Cursortasten> - bewegen des Cursors im Raster (12x21)

<Return> - aktuelles Sprite ablegen und nächstes Sprite wählen

< + >/<-> - Sprite wählen.

<**L**> - DOS-Command

...sendet ein Kommando zur Diskettenstation (s. Floppy-Handbuch). Damit wird es möglich, innerhalb des Programms Disketten zu formatieren etc.

<**F3**> **- modify (Grafik bearbeiten)**

...birgt zwei Grundfunktionen, Farben ändern und statistische Farbanalyse. Jeder Arbeitsschritt läßt sich mit <RUN/STOP> abbrechen. Nach dieser Unterbrechung springt das Programm zum Hauptmenü zurück.

<**A**> - Color Analysis

...ergibt eine Farbanalyse durch ein Balkendiagramm (Abb. 4). Es zeigt die Häufigkeit der verwendeten Farben. Die Balken sind mit der Farbnummer beschriftet (hexadezimal). Am unteren Bildschirmrand sind die einzelnen Farben mit ihren Kennwerten dargestellt.

<**B**> - Change Colors

...läßt eine Änderung der verwendeten Farben zu. Zuerst geben Sie die zu verändernde Farbe an (Farbnummer 0 bis F), danach die neue Farbnummer. Die Veränderung läßt sich sofort am Bildschirm beurteilen und ggf. korrigieren. <RUN/STOP> bringt Sie zurück zum Hauptmenü.

<**F5**> **- Output (Grafik ausgeben)**

Dieses Menü enthält Funktionen wie Directory, Diskbefehle, sowie das Speichern in den Formaten Amica Paint, Koalapainter, Adv. Art Studio und Blazing Paddles. Ebenso steht eine Hardcopyfunktion für Epson-kompatible Drucker zur Verfügung.

<**A**> - Epson Hardcopy

...druckt eine Graustufen-Hardcopy auf einem Epson-kompatiblen Printer. Die Frage nach Linefeed muß zuvor mit »Y«es oder »N«o beantwortet werden. Achten Sie hier auf die Stellung der DIL-Schalter in Ihrem Drucker.

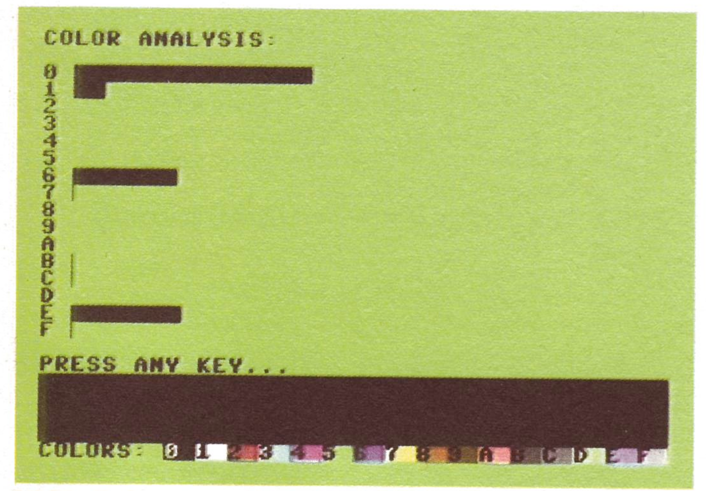
<**C**> - Amica-Paint

...speichert nach der Namenseingabe ein Bild im Amica-Paint-Format. Der Prefix (»[B]«) wird automatisch vorgestellt, läßt sich aber überschreiben.

<**E**> - Advanced Art Studio
...speichert nach der Namenseingabe ein Bild im Advanced-Art-Studio-Format. Der Suffix (»MPIC«) wird dabei automatisch hinter den Namen gestellt, läßt sich aber ebenfalls überschreiben.

<**G**> - Simple Bitmap
...speichert die Grafik im Multicolorstandard des VIC. Als erste Eingabe wird die Farbnummer für den Hintergrund erfragt, danach die Farbnummern der Bitmap-Kombinationen »01« und »10«. die vierte Farbe wird automatisch der Bitkombination »11« zugeordnet. Falls mehr als vier Farben vorkommen, erhält man eine Fehlermeldung und das Bild wird nicht gespeichert. Brechen Sie in so einem Fall mit <RUN/STOP> ab und vereinfachen Sie die Farbgebung (F3 - Modify). Die Bitmap erhält die Startadresse $4000.

<**I**> - char+video+color
... wandelt die gerade sichtbare Grafik in die Formate Zeichensatz, Video-RAM und Farb-RAM und speichert diese Daten nach Eingabe der Namen. Zeichensatzgrafik hat den Vorteil, weniger Speicherplatz zu belegen, und sich schneller verarbeiten zu lassen (z.B. beim Scrolling in Spielen). Das Bild kann dabei mehr als nur vier Farben enthalten. Bei der Farbeingabe wird wie bei »simple bitmap« verfahren. Das Bild selbst darf nicht zu komplex sein. Das heißt, es muß viele sich wiederholende 4*8 Pixel-Kästchen aufweisen, da nur 256 Zeichen zur Verfügung stehen, das ganze Bild jedoch aus 1000 Kästchen besteht. Der errechnete Zeichensatz hat noch eine Besonderheit: Character #0 wird in jedem Fall als Leerzeichen definiert, da diesem Zeichen in der Regel eine besondere Bedeutung zukommt. Der gespeicherte Zeichensatz ist immer 8*256 Bytes lang, egal wie viele Zeichen tatsächlich benötigt wurden. Der nicht verwendete Speicherbereich des Zeichensatzes wird mit Null-Bytes aufgefüllt. Beachten Sie, daß sich im Farb-RAM nur die Farben 0-7 verwenden lassen,



[4] Die Farbanalyse gibt Aufschluß über die Verteilung der einzelnen Farben im Bild

da Bit 3 als Flag für den Multicolormodus dient. Amica-Paint Graphics-Converter setzt dieses Bit automatisch. Ein Zeichensatz erhält die Startadresse $8000, das Video-RAM $C000. Das abgespeicherte Farb-RAM hat die Startadresse $C400.

<**K**> - Directory
...zeigt den Inhalt der aktuellen Diskette

<**B**> - Koalapainter
...die im Speicher befindliche Grafik wird im Koalapainter-Format (nach der Namenseingabe) gespeichert. Der Prefix »APIC« wird dabei automatisch vorgegeben. Beachten Sie, daß der erste Buchstabe ein reverses »A« ist. Er läßt sich, wenn Sie ihn einmal überschrieben oder gelöscht haben (<SHIFT CLR/HOME>) im Programm nicht mehr eintippen. Beenden Sie in diesem Fall mit <RUN/STOP>. Danach rufen Sie die Option neu auf (<F5>, danach <B>).

<**D**> - paint magic
...speichert im Paint-Magic-Format. Da Paint Magic die Farbmöglichkeiten des C64 nicht voll ausnutzt, ist eine komplette Speicherung manchmal nicht möglich. Im Paint Magic-Format ist die Farbe des Color-RAM ($D800) für das komplette Bild einheitlich gesetzt. Amica-Paint Graphics-Converter sucht bei der Konvertierung diejenige Farbe heraus, die am besten für das Color-RAM verwendet wird, und ordnet Video-RAM und Color-RAM entsprechend um. Falls dies nicht befriedigend gelingt, erhalten Sie eine Fehlermeldung und die Konvertierung wird unterbrochen. Die Stelle an der abgebrochen wurde wird mit einem 4x8-Pixel großen Kästchen angezeigt. Gehen Sie zurück zum Hauptmenü (mit <RUN/STOP>) und vereinfachen Sie (F3 - Modify).

<**F**> - Blazing Paddles
...speichert nach der Namenseingabe das Bild im Blazing-Paddles-Format. Der Prefix »PI.« wird dabei automatisch vorgestellt und läßt sich wieder überschreiben.

<**H**> - char+videoram
...wandelt Bitmap-Grafik in Zeichensatzgrafik und speichert beide. Die Grafik darf nur vier Farben enthalten. Nach dem Anwählen dieser Option fragt der Konverter nach dem Betriebsmodus: Um ein einzelnes Bild umzuwandeln, muß der Normalmodus (B) angewählt werden. Alternativ hierzu gibt es den Playfield-Modus. Dieser dient zum Umwandeln mehrerer Einzelbilder, wobei nur ein einziger Zeichensatz verwendet wird. Nachdem das erste Bild im Playfield-Modus verarbeitet wurde, besteht beim nächsten Aufruf dieser Option die Möglichkeit, das nächste (next) oder letzte (last) Bild des Playfields umzuwandeln, oder den Playfield-Modus zu beenden, wobei der schon berechnete Zeichensatz verloren geht. Nach dem Umrechnen des letzten Bildes wird der Zeichensatz gespeichert.

Ein Zeichensatz erhält die Startadresse $8000, das Videoram $C000.

<**J**> - Sprites
...erlaubt ein Speichern von einem oder mehreren Sprites. Die Option funktioniert weitestgehend wie das Laden. Auch hier kann man die zu speichernden Sprites automatisch (A) oder individuell (I) auswählen (lassen). Bei individueller Auswahl beendet <Pfeil nach links> die Zusammenstellung. Mit < + > und <-> kann man die schon gewählten Sprites betrachten. Falls beim Aufnehmen eines Sprites, egal in welchem Modus, innerhalb des Bildausschnitts Farben auftauchen, die bei der Farbeingabe nicht berücksichtigt wurden, bricht das Programm mit einer Fehlermeldung ab. Beim automatischen Aufnehmen wird nur der Bereich gespeichert, bis zu dem Grafikdaten vorhanden sind. Leere Spritedefinitionen werden ignoriert.

<**L**> - DOS-Command
...sendet ein Kommando zur Diskettenstation (s. Floppy-Handbuch). Damit wird es möglich, innerhalb des Programms Disketten zu formatieren etc.

<**F7**> - Watch
Dies ist die einfachste Programmfunktion. Sie blendet alle Menüs aus, und zeigt das komplette Bild. Zurück zum Hauptmenü geht's mit einer beliebigen Taste.

Mit Amica-Convert sind Sie den verschiedenen Speicher- und Kompressionsverfahren endlich gewachsen und können die Vorzüge der einzelnen Malprogramme ausschöpfen. (gr)

Kurzinfo: Amica-Convert

Programmart: Grafik-Converter
Laden: LOAD "AMICA-CONVERT",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 38
Programmautor: Oliver Stiller

Marke Eigenbau

Sie haben mit Sicherheit auch schon die farbigen Titel und Logos in Spielen bewundert. Ab jetzt ist es ein Kinderspiel, solche Vorspänne selbst zu basteln.

Es ist eine Tortur, eigene Vorspänne zu konstruieren: Entweder man zeichnet in einem Malprogramm ein kompliziertes Bild, lädt es (dauert ewig!) nach, oder einfache Blockgrafik hilft als Krücke über den Programmbeginn. Beide Variationen sind nicht gerade »das Gelbe vom Ei«. Wir schlagen Ihnen eine dritte, praktikablere Methode vor – den TMC-Editor. Mit ihm wird es endlich möglich, ohne Multicolor und ähnliche Tricks ein dreifarbiges Logo zu zeichnen. Dazu ist ein Joystick in Port 2 nötig, die Funktionen werden über die Tastatur aufgerufen. Ein Help-Menü gibt Ihnen jederzeit Auskunft über diesen Befehlssatz (<SHIFT H> Abb. 1).

Laden und Starten

Bevor Sie den Editor laden, sollten Sie alle Erweiterungen und Tools ausschalten, die den Raster-Interrupt beeinflussen. Danach laden Sie mit

```
LOAD"TMC-EDITOR",8
```

von der beiliegenden Diskette. Wundern Sie sich nicht, wenn der Cursor nach dem Programmstart (mit RUN) noch eine Weile blinkt. Der Editor initialisiert sich im IRQ. Danach wird der Bildschirm dunkel und ein Intro-Text erscheint. Er läßt sich mit <SPACE> verlassen. Unmittelbar danach sehen Sie das Help-Menü, das Sie mit beliebigem Tastendruck zum eigentlichen Editor hin verlassen können (Abb. 2). Von hier zeichnen Sie mit dem Joystick in Port 2 und erreichen die anderen Funktionen über die Tastatur:

Cursor-Position

Der Bildschirm ist zweigeteilt. In der oberen Hälfte wird das gesamte Logo dargestellt (256 x 84 Punkte), in der unteren ein stark vergrößerter Ausschnitt. Hier befindet sich auch der Cursor, den Sie mit einem in Port 2 eingestöpselten Joystick bewegen können. Wird er zu den Endbereichen des Fensters bewegt, scrollt der Bildschirminhalt. Zum Sichtbarmachen der Position auf der Gesamtdarstellung gibt es eine Sonderfunktion.

<SPACE> – macht die aktuelle Cursor-Position als Strich am Gesamtbildschirm (obere Hälfte) sichtbar. Jede Aktion von Joystick oder Tastatur schaltet diese Funktion wieder aus.

Zeichenfunktionen

Normalerweise zeichnet ein Druck des Feuerknopfes (<Feuer>) einen Punkt in der gewählten Farbe (s. Farbwahl) unter den Cursor. Als Ausnahme von der Regel gilt die Kopierfunktion. Hier markiert oder kopiert <Feuer> (s.d.). Zusätzlich sollten Sie beachten, daß in einer Zeile (acht Punkte innerhalb des Gitters) nur zwei Farben erlaubt sind. Wenn Sie hier eine Farbe ändern, nehmen alle Punkte der alten Color die neue an. Welche tolle Farbkombinationen damit möglich sind, zeigt das Demo nach dem Programmstart (Abb. 2). Um es zu löschen, drücken Sie:

<CLR/HOME> – löscht den gesamten Bildschirm.

Markierungsfarben

Die Grundfarben von Cursor, Hilfsgitter und Hintergrund sind bei entsprechender Farbsetzung unsichtbar und lassen sich daher ändern. Auch hier färbt sich der Rahmen mit. Jeder

```
                HELP-MENU

  KEY:                FUNCTION:

          1-8       -CHOOSE COLOUR 1-8
 <SHIFT>  1-8       -CHOOSE COLOUR 9-16
          F1        -COPY
          F3        -STOP COPY

          F5        -CHANGE CURSOR COLOUR
          F6        -CHANGE GRID COLOUR
          F7        -CHANGE BACKGROUND COLOUR

 <SPACE>            -SHOWS POSITION
 <CLR/HOME>         -CLEARS PICCY

 <SHIFT> D          -DIRECTORY
 <SHIFT> L          -LOAD PICCY
 <SHIFT> S          -SAVE PICCY
 <SHIFT> H          -SHOWS HELP-MENU

 <RUN/STOP>         -RETURN TO DRAW-MODE

     -< USE JOYSTICK II TO DRAW >-
```

[1] Das Help-Menü zeigt die Tastaturbelegung

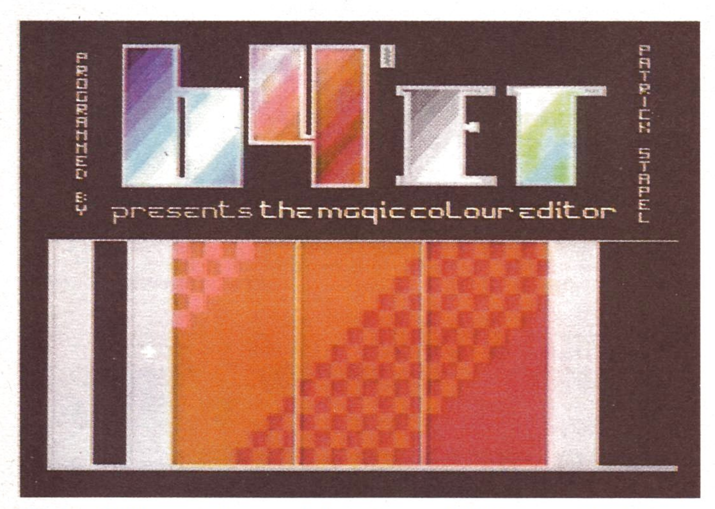
entsprechende Tastendruck (<F5>, <F6> und <F7>) ändert die Farbnummer.
<F5> - ändert Cursor-Farbe.
<F6> - wechselt Gitterfarbe
<F7> - wählt andere Hintergrundfarbe

Farbwahl

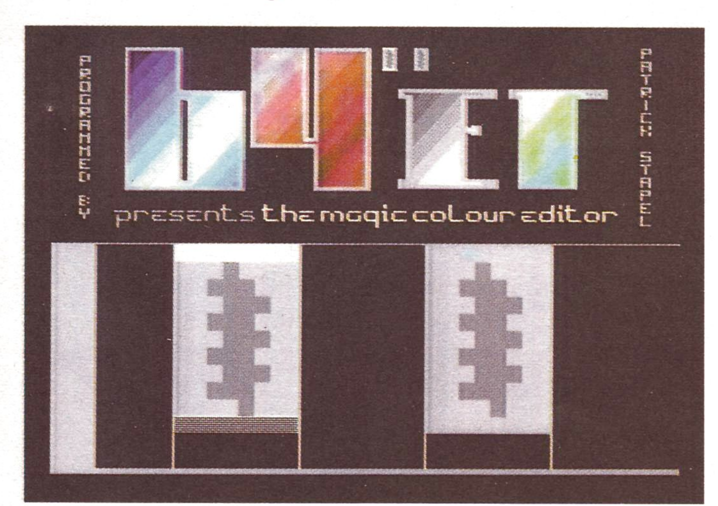
Die einzelnen Farben wählen Sie über die Tasten <1> bis <8> und <SHIFT 1> bis <SHIFT 8>. Zur Verdeutlichung färbt sich der Bildschirmrahmen entsprechend. Danach werden alle Zeichenfunktionen in dieser Farbe ausgeführt.

<1> bis <8>
...entspricht den Farbnummern 0 bis 7 des normalen C-64-Modus:
<1> - Schwarz
<2> - Weiß
<3> - Rot
<4> - Cyan
<5> - Purpur
<6> - Grün
<7> - Blau
<8> - Gelb

<SHIFT 1> bis <SHIFT 8>
...entspricht den Farbnummern 8 bis 15:
<SHIFT 1> - Orange
<SHIFT 2> - Braun
<SHIFT 3> - Hellrot
<SHIFT 4> - Dunkelgrau
<SHIFT 5> - Mittelgrau
<SHIFT 6> - Hellgrün
<SHIFT 7> - Hellblau
<SHIFT 8> - Hellgrau



[2] Der Editor zeigt einmal das Gesamtbild (obere Hälfte) und einen Bearbeitungsausschnitt



[3] Bei der Kopierfunktion markiert je ein blinkender Strich den gewünschten Block links oben und rechts unten

Kopierfunktion

Eine komfortable Block-Copy-Funktion ermöglicht ein Vervielfältigen beliebiger Bereiche. Allerdings läßt sich jeweils nur ein Block von mindestens acht Punkten Breite und zwei Punkten Höhe kopieren. Eingeleitet wird der Befehl mit <F1> und beendet, bzw. unterbrochen mit <F3>. Sie legen zuerst die linke obere Kante fest (Joystick-<Feuer>), dann die rechte untere. Die markierten Start- und Endkanten blinken (Abb. 3). Jedes weitere <Feuer> kopiert den Bereich ab Spaltenbeginn der Cursor-Position:

<F1>
...schaltet den Copy-Modus ein.

<F3>
...schaltet den Copy-Modus aus.

<Feuer>
...markiert oder kopiert im Copy-Modus die gewünschten Felder.

Diskettenoperationen

Um das Programm so einfach wie möglich zu machen, wurden nur die wichtigsten Funktionen für Diskettenoperationen mit eingebaut. Eingabefehler werden abgefangen, allerdings führt ein nicht vorhandenes File auf Diskette oder ein fehlerhafter Ladevorgang zu falscher Bildschirmausgabe. Die Operationen zum Laden bzw. Speichern lassen sich mit <RUN/STOP> unterbrechen. Vor die einzelnen Files wird automatisch ein Prefix (»->«) gestellt, da der TMC-Editor ein eigenes Datenformat besitzt.

<SHIFT D>
...zeigt das Inhaltsverzeichnis einer eingelegten Diskette.

<SHIFT L>
...lädt ein TMC-Bild nach Eingabe des Programmnamens. Der Prefix wird automatisch vorangestellt.

<SHIFT S>
...speichert ein TMC-Bild nach Eingabe des Programmnamens. Der Prefix wird automatisch vorangestellt und braucht nicht eingetippt werden.

Datenfiles

Die erzeugten Files sind bootfähig. Das heißt, wenn Sie beispielsweise ein Programm unter dem Namen »TEST« gespeichert haben und mit:

```
LOAD"->TEST",8
```

außerhalb des TMC-Editors laden, läßt es sich mit RUN starten. Am Bildschirm erscheint dann Ihr eigenes Logo. Ein beliebiger Tastendruck führt weiter. (gr)

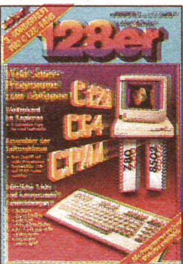
Kurzinfo: TMC-Editor

Programmart: Logomaker
Laden: LOAD"TMC-EDITOR",8
Starten: nach dem Laden RUN eingeben
Steuerung: Joystick in Port 2 und Tastatur
Besonderheiten: Die Bild-Files lassen sich mit RUN starten und betrachten
Benötigte Blocks: 38
Programmautor: Patrick Stapel

64'er Sonderhefte alle auf einen Blick

Die 64'er Sonderhefte bieten Ihnen umfassende Information in komprimierter Form zu speziellen Themen rund um die Commodore C 64 und C 128. Ausgaben, die eine Diskette enthalten, sind mit einem Diskettensymbol gekennzeichnet.

C 64, C 128, EINSTEIGER



SH 0022: C 128 III
Farbiges Scrolling im 80-Zeichen Modus / 8-Sekunden-Kopierprogramm



SH 0026: Rund um den C64
Der C64 verständlich für Alle mit ausführlichen Kursen



SH 0029: C 128
Starke Software für C 128/ C 128D / Alles über den neuen C 128D im Blechgehäuse





SH 0036: C 128
Power 128: Directory komfortabel organisieren / Haushaltsbuch: Finanzen im Griff / 3D-Landschaften auf dem Computer



SH 0038: Einsteiger
Alles für den leichten Einstieg / Super Malprogramm / Tolles Spiel zum selbermachen / Mehr Spaß am Lernen

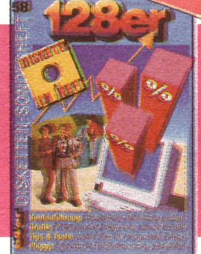




SH 0050: Starthilfe
Alles für den leichten Einstieg / Heiße Rythmen mit dem C 64 / Fantastisches Malprogramm



SH 0051: C 128
Volle Floppy-Power mit "Rubikon" / Aktienverwaltung mit "Börse 128"



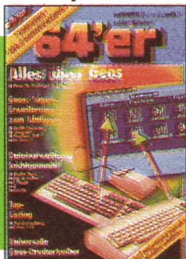
SH 0058: 128er
Übersichtliche Buchhaltung zuhause / Professionelle Diagramme



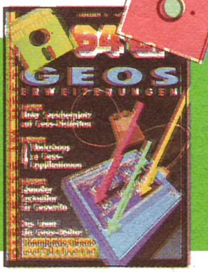
SH 0062: Erste Schritte
RAM-Exos: Disketten superschnell geladen/ Exbasic Level II: über 70 neue Befehle/ Rafinessen mit der Tastatur

SH 0064: 128ER
Anwendungen: USA Journal / Grundlagen: CP/M, das dritte Betriebssystem/ VDC-Grafik: Vorhang auf für hohe Auflösung

GEOS, DATEIVERWALTUNG



SH 0028: Geos / Dateiverwaltung
Viele Kurse zu Geos / Tolle Geos-Programme zum Abtippen



SH 0048: GEOS
Mehr Speicherplatz auf Geos-Disketten / Schneller Texteditor für Geowrite / Komplettes Demo auf Diskette



SH 0059: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks

PROGRAMMIERSPRACHEN



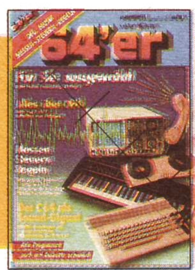


SH 0035: Assembler
Abgeschlossene Kurse für Anfänger und Fortgeschrittene



SH 0040: Basic
Basic Schritt für Schritt / Keine Chance für Fehler / Profi-Tools und viele Tips

ANWENDUNGEN



SH 0031: DFÜ, Musik, Messen-Steuern-Regeln
Alles über DFÜ / BTX von A-Z / Grundlagen / Bauanleitungen



SH 0046: Anwendungen
Das erste Expertensystem für den C 64 / Bessere Noten in Chemie / Komfortable Dateiverwaltung



SH 0056: Anwendungen
Gewinnauswertung beim Systemlotto / Energieverbrauch voll im Griff / Höhere Mathematik und C64

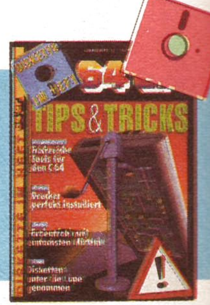
TIPS, TRICKS & TOOLS



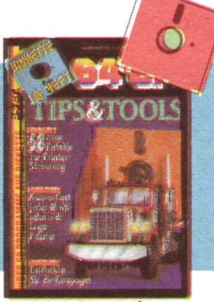
SH 0024: Tips, Tricks& Tools
Die besten Peeks und Pokes sowie Utilities mit Pfiff



SH 0043: Tips, Tricks&Tools
Rasterinterrupts - nicht nur für Profis / Checksummer V3 und MSE / Programmierhilfen



SH 0057: Tips & Tricks
Trickreiche Tools für den C64 / Drucker perfekt installiert



SH 0065: Tips&Tools
Streifzug dirch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks

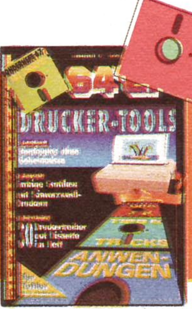
HARDWARE



SH 0025: Floppylaufwerke
Wertvolle Tips und Informationen für Einsteiger und Fortgeschrittene



SH 0032: Floppylaufwerke und Drucker
Tips&Tools / RAM-Erweiterung des C64 / Druckerroutinen



SH 0047: Drucker, Tools
Hardcopies ohne Geheinmisse / Farbige Grafiken auf s/w-Druckern



SH 0067: Wetterstation:
Temperatur, Luftdruck und feuchte messen/ DCF-Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC

DTP



SH 0039: DTP, Textverarbeitung
Komplettes DTP-Paket zum Abtippen / Super Textsystem / Hochauflösendes Zeichenprogramm

64'er Magazin auf einen Blick

Diese 64'er-Ausgaben bekommen Sie noch bei Markt&Technik für jeweils 7,- DM. Die Preise für Sonderhefte und Sammelbox entnehmen Sie bitte dem Bestellcoupon. Tragen Sie Ihre Bestellung im Coupon ein und schicken Sie ihn am besten gleich los, oder rufen Sie einfach unter 089 - 20 25 15 28 an.

11/90: Bausatztest: Der Taschengeldplotter / Vergleichstest: Drucker der Spitzenklasse / 5 Schnellbauschaltungen

12/90: Abenteuer BTX / Multitasking für C64 / Großer Spieleschwerpunkt / Programmierwettbewerb: 30 000 DM zu gewinnen

1/91: Die Besten Tips&Tricks / Neu: Reparaturecke / Floppy-Flop: Betriebssystem überlistet / Jahresinhaltsverzeichnis

2/91: Sensation: Festplatte für den C 64 / Drucken ohne Ärger / Listing des Monats: Actionspiel "Ignition" / Longplay: Dragon Wars

3/91: Bauanleitung: Universelles Track-Display / Alles über Module für den C 64 / Festplatte HD 20 unter GEOS

4/91: Spiele-Schwerpunkt: 100 Tips, News, Tests / Neu: Grafikkurs / Fischer-Baukästen / Bauanleitung: Digitizer

5/91: Ätzanlage unter 50.- DM / GRB-Monitor am C64 / Longplay: Bard's Tale / Reparaturkurs: Die neuen C64 / Piratenknacker

6/91: C64er-Meßlabor: Universell erweiterungsfähig / Test: Pocket-Wrighter 3.0 - Bestes C64 Textprogramm / Listing des Monats: Autokosten im Griff

7/91: Trickfilm mit dem C 64 / Bauanleitung: 1541-Floppy mit Batteriebetrieb / Listing des Monats: Basic-Butler

8/91: Drucker unter 1000 DM / Test: GEO-RAM / Listing des Monats: 80-Farben-Malprogramm / Longplay: Secret of the Silver Plate

09/91: Joystick im Test / Die üblen Tricks mit Raubkopien / Die besten Drucker unter 1500 DM / Mit großem Spieleteil

10/91: 100 besten Tips&Tricks / Listing: Fraktal-Programm / C-64-Meßlabor: komfortables Kontrollmodul

11/91: Alles über Diskette & Floppy / Bauanleitung: C 64 steuert Laserstrahl / Sha-Jongg: Topspiel mit Spitzengrafik / Großer Spieleteil

12/91: Alle Spiele 1991 / Tolle Tips&Tricks für den C64 und C128 / Geschenktips für Computerfans / Komfortable Videoverwaltung

GRAFIK



SH 0020: Grafik
Grafik-Programmierung / Bewegungen



SH 0045: Grafik
Listings mit Pfiff / Alles über Grafik-Programmierung / Erweiterungen für Amica-Paint



SH 0055: Grafik
Amica-Paint: Malen wie ein Profi / DTP-Seiten vom C64 / Tricks&Utilities zur Hires-Grafik



SH 0063: Grafik
Text und Grafik mischen ohne Flimmern / EGA: Zeichenprogramm der Superlative / 3 professionelle Editoren



SH 0068: Anwendungen
Kreuzworträtsel selbstgemacht/ Happy Synth: Super-Synthesizer für Sound-Freaks/ Der C64 wird zum Planetarium/ Sir-Compact: Bit-Packer verdichtet Basic- und Assemblerprogramme

SPIELE



SH 0030: Spiele für C 64 und C 128
Spiele zum Abtippen für C 64/ C 128 / Spieleprogrammierung



SH 0037: Spiele
Adventure, Action, Geschicklichkeit / Profihilfen für Spiele / Überblick, Tips zum Spielekauf



SH 0042: Spiele
Profispiele selbst gemacht / Adventure, Action, Strategie



SH 0049: Spiele
Action, Adventure, Strategie / Sprites selbst erstellen / Virenkiller gegen verseuchte Disketten



SH 0052: Abenteuerspiele
Selbstprogrammieren: Von der Idee zum fertigen Spiel / So knacken Sie Adventures



SH 0054: Spiele
15 tolle Spiele auf Diskette/ der Sieger unseres Programmierwettbewerbs: Crillion II/ ein Cracker packt aus: ewige Leben bei kommerziellen Spielen



SH 0060: Adventures
8 Reisen ins Land der Fantasie - so macht Spannung Spaß



Top Spiele 1
Die 111 besten Spiele im Test/ Tips, Tricks und Kniffe zu heißen Games/ Komplettlösung zu "Last Ninja II"/ große Marktübersicht: die aktuellen Superspiele für den C64



SH 0061: Spiele
20 heiße Super Games für Joystick-Akrobaten/ Cheat-Modi und Trainer POKEs zu über 20 Profi-Spielen/ Krieg der Kerne: Grundlagen zur Spielerprogrammierung



SH 0066: Spiele
15 Top-Spiele mit Action und Strategie/ Mondlandung: verblüffend echte Simulation und Super-Grafik/ High-Score-Knacker: Tips&Tricks zu Action-Games

BESTELLCOUPON

Ich bestelle ____ 64er Sonderhefte Nr. ________________ _____ DM
zum Preis von je: 14,- DM (Heft ohne Diskette) _____ DM
16,- DM (Heft mit Diskette) _____ DM
9,80 DM (SH "Top Spiele 1") _____ DM
24,- DM (für die Sonderhefte 0051/ 0058/ 0064) _____ DM

Ich bestelle ____ 64er Magazin Nr. ___/ ___/ ___ _____ DM
zum Preis von je 7,- DM

Ich bestelle ____ Sammelbox(en) _____ DM
zum Preis von je 14,- DM

Gesamtbetrag ______ DM

Ich bezahle den Gesamtbetrag zzgl. Versandkosten nach Erhalt der Rechnung.

Name, Vorname

Straße, Hausnummer

PLZ, Wohnort

Telefon (Vorwahl) ______________ Ich erlaube Ihnen hiermit mir interessante Zeitschriftenangebote auch telefonisch zu unterbreiten (ggf. streichen).

Schicken Sie bitte den ausgefüllten Bestellcoupon an: **64er Leserservice, CSJ, Postfach 140 220, 8000 München 5, Telefon 089/ 20 25 15 28**

Praktisches Kleinzeug

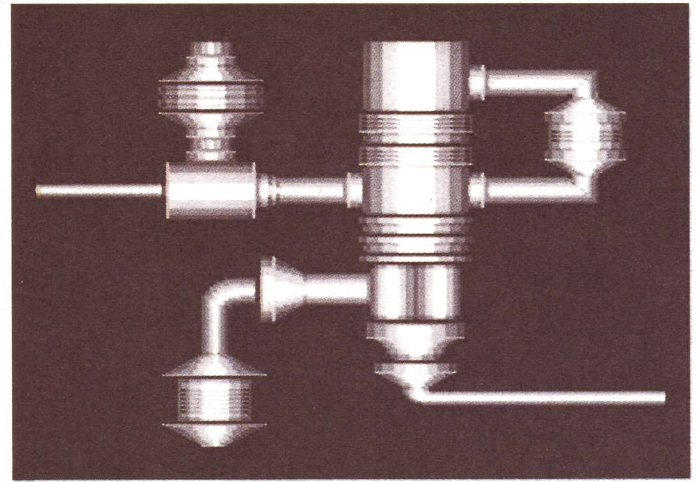
»Stamp-Maker« definiert und verwaltet Ausschnitte aus Grafiken bis zu einer Maximalgröße von 25 x 31 Cursor-Punkten – Hires-Windows, Animationen und zur Verwaltung von Kleinbildern (Minidiashow).

Nicht nur Programmierer werden von diesem Tool begeistert sein, generiert und verwaltet es doch Ausschnitte (Stamps) aus Grafiken. Eine Vielzahl von Anwendungen werden zum Kinderspiel. Beispielsweise benötigen Animationen extrem viel Speicher; eine einzige Grafik (ohne Farbspeicher) schon 8 KByte. Bei simulierter Bewegung werden mehrere Bilder (mit jeweils kleinen Unterschieden) nacheinander auf dem Bildschirm gezeigt. Es leuchtet ein, daß hier die 64 KByte des C64 sehr schnell belegt sind. Der Stamp-Maker umgeht diesen Aufwand:

Sie zeichnen zuerst in einem Malprogramm einen Prototyp. Dann verändern Sie dieses Bild in mehreren Stufen, bis eine Animationssequenz entstanden ist. Natürlich muß jeder Zwischenstand gespeichert werden. Im zweiten Schritt werden die unterschiedlichen Gemälde geladen und mit Stamp-Maker nur die veränderten Bildanteile gespeichert. Nun die Datensätze zusammenfassen (z.B. mit einem Monitor nacheinander in den Speicher laden) und zusammen mit dem ersten Bild speichern. Bei der Vorführung lädt man zuerst Ursprungsbild und Stamps. Das Ursprungsbild wird auf den Bildschirm gebracht und Stamp-Maker kopiert nacheinander die Veränderungen hinein. Das Ergebnis ist eine Animation, die weniger Speicher als mit kompletten Grafiken verschwendet und daher wesentlich komplexer sein kann.

Aber auch Windows lassen sich relativ einfach verwenden, z.B. beim Vorführen Ihrer Grafiken:

Sie konstruieren in einem Malprogramm einige Icons (Symbole für Funktionen) in einem Bild. Danach definieren Sie diese als Stamps. Es bleibt nur noch eine Nachladeroutine für Ihre Grafiken zu schreiben, den Bildschirm auf diese Darstellungsart umzuschalten und per Joystick einzelne Windows einzublenden – dafür genügt das eingebaute Basic des C64. Die Routinen lassen sich sehr schnell programmieren und sind trotzdem professionell.



Im Demo sorgen Stamps für Bewegung

Damit Sie eine Ahnung von der Leistungsfähigkeit bekommen, befindet sich auf der beiliegenden Diskette ein Basic-Demoprogramm. Geladen wird es mit

```
LOAD"STAMP-MAKER-DEMO",8
```

und gestartet mit RUN. Danach wird zuerst »Stamp-Maker V2« und die Files »Demo-1«, »Demo-2« und »Demo-3« nachgeladen. Im Anschluß daran erhalten Sie einen Infotext, der mit <SPACE> abgebrochen wird – eine Animationsgrafik wird sichtbar (Abb.). Wenn Sie mit <RUN/STOP RESTORE> abbrechen, empfiehlt es sich via LIST das Programm näher zu betrachten. Hier sehen Sie die Verwendung aller Befehle.

Wie Sie aus dem Demo erkennen, ist der Stamp-Maker als Betriebssystemerweiterung ausgelegt. Geladen wird er mit

```
LOAD"STAMP-MAKER V2",8,1
```

Danach sollten Sie NEW eingeben. Achtung: Es ist zwar möglich, im Direktmodus auf den Befehlssatz zuzugreifen, aber sinnvoller ist die Abarbeitung in einem speziell für Ihren Bedarfszweck angepaßten Programm. Dazu ist eine Nachladeroutine erforderlich, wie auch im Demo verwendet:

```
5 IF A=0 THEN A=1:LOAD"STAMP-MAKER V2",8,1
```

Anschließend stehen die Befehle im Programm zur Verfügung:

Ausschnitt bearbeiten

SYS49152,x1,y1,b1,z1,c1,b2,z2,c2,ro,pa,x2,y2

... ist der mächtigste, aber auch komplexeste Befehl. Er ermöglicht sowohl das Erzeugen als auch das Setzen eines Grafikausschnitts. Erlaubt sind dabei Ausschnitte bis zur Größe von 25 x 31 Cursor-Punkten. Der Cursor hat hier eine Breite von acht Pixel und eine Höhe von einem Pixel. Dabei sind die Parameter von Bedeutung:

x1 – x-Wert der Größe des Grafikausschnitts
y1 – y-Wert der Größe des Grafikausschnitts
b1 – Startadresse der zu lesenden Bitmap
z1 – Startadresse des zu lesenden Video-RAM
c1 – Startadresse des zu lesenden Color-RAM
b2 – Startadresse des zu schreibenden Bereichs
z2 – Startadresse des zu schreibenden Video-RAM
c2 – Startadresse des zu schreibenden Color-RAM
ro – Lesebereich = unter ROM (1) oder im normalen AM (0)
pa – gibt an, ob der zu schreibende Bereich gelesen oder geschrieben werden soll (»1« = Stamp aus Bitmap konvertieren, »0« = Stamp in Bitmap-Format wandeln und schreiben). Ist pa=1 werden b1,z1, und c1 aus der Bitmap als Stamp umgewandelt und in diesem Format nach b2,z2 und c2 geschrieben. Wenn pa=0, wird ein Stamp aus b1, z1 und c1 gelesen ins Bitmap-Format gewandelt und nach b2, z2 und c2 geschrieben.

Zum Herausschneiden eines Bereichs müssen b1, z1 und c1 die Startadressen des Bildes enthalten und b2, z2 und c2 die Startadressen des Speicherbereichs, in den das Stamp soll (pa=1). Will man es auslesen und auf eine Grafikseite setzen, muß pa=0 sein. b1, z1, und c1 müssen dann die Startadressen der Grafikseite enthalten.

x2 – x-Position der Grafikseite
y2 – y-Position der Grafikseite

Dazu zwei Beispiele:

Stamp erzeugen (Achtung: Die Zeile muß fortlaufend eingegeben werden, ein Zeilenumbruch ergibt sich am Bildschirm an einer anderen Stelle).

```
SYS49152,5,6,24576,32576,33576,57344,57584,57614,0
,1,0,0
```

... erzeugt ein Stamp der Größe x=5, y=6 ab Zeile 0 und Spalte 0 aus einer Bitmap ab Position 24576 und schreibt es nach 57344. Die Positionen für z2 (57584) und c2 (57614) müssen von Ihnen berechnet werden. Dabei spielt die Größe des Stamps eine Rolle:

Startadresse Hires: 57344
Startadresse Video-RAM: (5x8)x6+57344 = 57584
Startadresse Color-RAM: (5x6)+57584 = 57614
Endadresse = nächste Startadresse: (5x6)+57614 = 57644

Stamp auf eine Grafikseite setzen (Achtung: Die Zeile muß fortlaufend eingegeben werden, ein Zeilenumbruch ergibt sich am Bildschirm an einer anderen Stelle).

```
SYS49152,5,6,57344,57584,57614,8192,1024,55296,1,0
,8,9
```

... bringt das im ersten Beispiel erzeugte Stamp ab Spalte 8 (Pixel 8x8=64) und Zeile 9 in eine Grafikmap ab Position 8192. Die Startadressen des Video- und Color-RAM richten sich dabei nach der Grafikseite (Video=1024, Color=55296). Die Umschaltung ro=1 ist entsprechend, da ROM-Bereich.

Speichern auf Floppy

SYS49910,Startadresse,Endadresse,"Filename"

... speichert einen Bereich zwischen »Startadresse« und »Endadresse« unter dem Namen »Filename« auf Diskette.

Laden von Floppy

SYS50033,"Filename"

... lädt ein PRG-File von Diskette an die definierte Speicherposition. Dieser Befehl entspricht weitgehend »LOAD"Filename",8,1«. Eine Eingabe von NEW ist nicht notwendig (s.a. Zeile 540 bis 542 im Demoprogramm).

Bearbeiten von Speicherbereichen

SYS50075,Startadresse1,Startadresse2

... verschiebt Video-RAM oder Color-RAM von Startadresse1 nach Startadresse2. Dabei ist die Länge fest auf 1024 Bytes festgelegt.

```
SYS 50075,32576,23552:REM VIDEO-RAM
SYS 50075,33576,55296:REM COLOR-RAM
```

... verschiebt Video- und Farbspeicher eines am Originalspeicherplatz geladenen Koala-Paint-Bildes.

SYS50135,Startadresse1

... löscht eine Grafikseite. Die Anfangsadresse muß in Startadresse angegeben werden. Die Länge der Grafik ist auf 8192 Bytes festgelegt:

```
SYS50135,24576
```

... löscht eine Koala-Grafik.

Mit dem Stamp-Maker, sind nicht nur Kleindias machbar, sondern es gelingen Ihnen nach kurzer Einarbeitungszeit die tollsten Animationen. (gr)

Kurzinfo: Stamp-Maker

Programmart: Grafik-Tool
Laden: LOAD"STAMP-MAKER V2",8,1
Starten: nach dem Laden NEW eingeben, SYS49152 + Parameter
Besonderheiten: eingebauter Packmodus
Benötigte Blocks: 4
Programmautor: Jörg Neumann

Kurzinfo: Stamp-Maker-Demo

Programmart: Grafikdemo
Laden: LOAD"STAMP-MAKER-DEMO",8
Starten: nach dem Laden RUN
Besonderheiten: lädt die Files »Stamp-Maker V2« und »Demo-1 bis Demo-3< nach
Benötigte Blocks: 16
Programmautor: Jörg Neumann

Grafikdemos - tolle Bilder

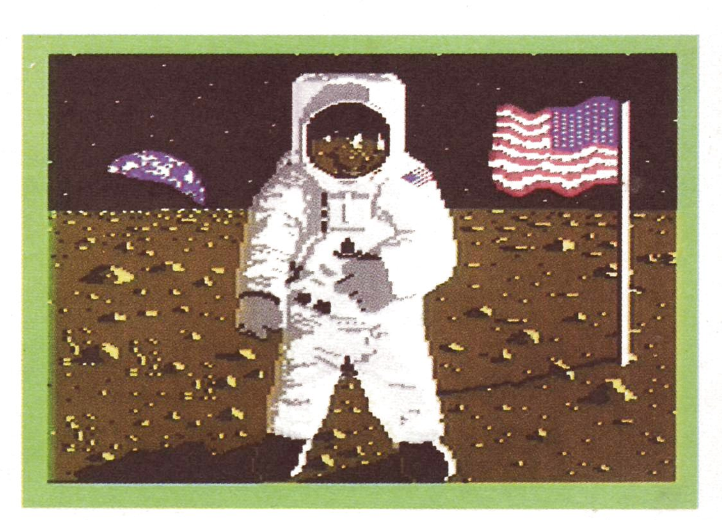
Faszinierende Einblicke

Zum Ausprobieren der Programme Amica-Convert und Stamp-Maker in diesem Heft finden Sie 16 ausgefeilte Grafiken in den unterschiedlichsten Formaten auf der beiliegenden Diskette.

Sie werden überrascht sein, welche Farbvielfalt auf dem C64 möglich ist (Abb.). Laden und betrachten lassen sich die Grafiken in Amica-Convert (S. 9) und konvertieren Sie die einzelnen Formate in Koala-Print. Dann lassen sich im Stamp-Maker einzelne Minibilder entnehmen und animieren - doch sehen Sie selbst. (gr)

Kurzinfo: Grafikdemobilder

Programmart: Multicolorbilder
Laden und betrachten: mit Amica-Convert
Benötigte Blocks: jeweils zwischen 40 und 41



Eines von den 16 Bildern der Heftdiskette

So finden Sie die Programme auf der Diskette

DISKETTE SEITE 1

```
0    "64'ER SONDERH.77" S1 2A
21   "DISKLADER"          PRG  S. 21
0    "┌────────────────┐" DEL
0    "|     GRAFIK     |" DEL
0    "└────────────────┘" DEL
38   "TMC-EDITOR"         PRG  S. 12
0    "------------------" DEL
38   "AMICA-CONVERT"      PRG  S. 9
0    "------------------" DEL
4    "STAMP-MAKER V2"     PRG  S. 16
16   "STAMP-MAKER-DEMO"   PRG
40   "DEMO-1"             PRG
32   "DEMO-2"             PRG
11   "DEMO-3"             PRG
0    "┌────────────────┐" DEL
0    "|     SOUND      |" DEL
0    "└────────────────┘" DEL
```

```
25   "R.C.S."             PRG  S. 4
3    "AXEL F."            PRG
11   "AXEL F./I"          PRG
3    "NINETEEN"           PRG
11   "NINE/I"             PRG
7    "RPS-DEMO"           PRG
7    "ORIGINAL-RHYTHME"   PRG
7    "PROGRESS-DEMO"      PRG
0    "------------------" DEL
30   "GITARRENSCHULE"     PRG  S. 8
45   "SOUND2"             PRG
0    "┌────────────────┐" DEL
0    "|     FLOPPY     |" DEL
0    "└────────────────┘" DEL
15   "REL-DEMOPROGRAMM"   PRG  S. 38
0    "------------------" DEL
22   "DISC-BASIC"         PRG  S. 44
```

```
0    "------------------" DEL
14   "ARC 1.5"            PRG  S. 42
0    "┌────────────────┐" DEL
0    "|  GRAFIKBILDER  |" DEL
0    "└────────────────┘" DEL
41   "PI.TAROT1"          PRG  S. 17
40   "TAROT2       MPIC"  PRG
40   "APIC ASTRONAUT  "   PRG
41   "PI.MANDRILL"        PRG
40   "HARDWARE     MPIC"  PRG
40   "APIC KNOTEN     "   PRG
0    "┌────────────────┐" DEL
0    "|    DISKETTE    |" DEL
0    "|   BEIDSEITIG   |" DEL
0    "|    BESPIELT    |" DEL
0    "└────────────────┘" DEL
22 BLOCKS FREE.
```

DISKETTE SEITE 2

```
0    "64'ER SONDERH.77" S2 2A
21   "DISKLADER"          PRG  S. 21
0    "┌────────────────┐" DEL
0    "|    DRUCKER     |" DEL
0    "└────────────────┘" DEL
8    "TABULA PRINT"       PRG  S. 37
8    "TABULA PRINT PAR"   PRG
12   "TR/BEISPIEL"        PRG
8    "TR/BLOCKDEMO"       PRG
3    "TR/TESTDATEI"       PRG
2    "TR/ITERATION"       PRG
0    "┌────────────────┐" DEL
0    "|    MONITOR     |" DEL
0    "└────────────────┘" DEL
29   "TESTBILD V2.1"      PRG  S. 49
0    "┌────────────────┐" DEL
0    "| PROGRAMMIEREN  |" DEL
0    "└────────────────┘" DEL
25   "SCHRITTSIMO"        PRG  S. 22
```

```
0    "------------------" DEL
26   "DECOMPILER"         PRG  S. 25
0    "------------------" DEL
31   "EXTEDITOR"          PRG  S. 26
0    "┌────────────────┐" DEL
0    "|    HARDWARE    |" DEL
0    "└────────────────┘" DEL
27   "HYP.PACKED"         PRG  S. 34
8    "INSTALL"            PRG
0    "------------------" DEL
34   "MG VERSION 1.0"     PRG  S. 32
0    "┌────────────────┐" DEL
0    "| TIPS & TRICKS  |" DEL  S. 29
0    "└────────────────┘" DEL
1    "CLIP"               PRG
2    "TOKEN-FINDER"       PRG
4    "ZEILENMARKER"       PRG
2    "RESET-SCHUTZ"       PRG
0    "------------------" DEL
```

```
5    "EINGABEGERAETE"     PRG
2    "INPUT.PEN"          PRG
0    "┌────────────────┐" DEL
0    "|  GRAFIKBILDER  |" DEL  S. 17
0    "└────────────────┘" DEL
40   "APIC SCHATTEN   "   PRG
41   "PI.ANTIRIAD"        PRG
40   "HARLEKIN     MPIC"  PRG
40   "APIC LUCKY LUKE"    PRG
41   "PI.SCHIFF"          PRG
40   "FOREST LIVE MPIC"   PRG
40   "APIC FORMEL1    "   PRG
41   "PI.ROSE"            PRG
40   "APIC AMERIKA    "   PRG
40   "MOUSE        MPIC"  PRG
0    "┌────────────────┐" DEL
0    "|      ENDE      |" DEL
0    "└────────────────┘" DEL
3 BLOCKS FREE.
```

WICHTIGE HINWEISE
zur beiliegenden Diskette:

Aus den Erfahrungen der bisherigen Sonderhefte mit Diskette wollen wir ein paar Tips an Sie weitergeben:

1 **Bevor Sie mit den Programmen auf der Diskette arbeiten, sollten Sie unbedingt eine Sicherheitskopie der Diskette anlegen. Verwenden Sie dazu ein beliebiges Kopierprogramm, das eine komplette Diskettenseite dupliziert.**

2 **Auf der Originaldiskette ist wegen der umfangreichen Programme nur wenig Speicherplatz frei. Dies führt bei den Anwendungen, die Daten auf die Diskette speichern, zu Speicherplatz-Problemen. Kopieren Sie daher das Programm, mit dem Sie arbeiten wollen, mit einem File-Copy-Programm auf eine leere, formatierte Diskette und nutzen Sie diese als Arbeitsdiskette.**

3 **Die Rückseite der Originaldiskette ist schreibgeschützt. Wenn Sie auf dieser Seite speichern wollen, müssen Sie vorher mit einem Diskettenlocher eine Kerbe an der linken oberen Seite der Diskette anbringen, um den Schreibschutz zu entfernen. Probleme lassen sich von vornherein vermeiden, wenn Sie die Hinweise unter Punkt 2 beachten.**

ALLE PROGRAMME
aus diesem Heft



HIER

Herausgeber: Carl-Franz von Quadt, Otmar Weber
Redaktionsdirektor: Dr. Manfred Gindle

Chefredakteur: Georg Klinge (gk) – verantwortlich für den redaktionellen Teil
Stellv. Chefredakteur: Arnd Wängler (aw)
Textchef: Jens Maasberg
Redaktion: Harald Beiler (bl), Herbert Großer (gr)
Produktion: Andrea Pfliegensdörfer
Redaktionsassistenz: Sylvia Wilhelm, Birgit Misera

So erreichen Sie die Redaktion:
Tel. 089/46 13-202, Telefax: 089/46 13-5001, Btx: 64064

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß die Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Dagmar Portugall
Bildredaktion: Wallo Linne (Ltg.), Roland Müller, Tina Steiner (Fotografie), Werner Nienstedt (Computergrafik)

Anzeigendirektion: Jens Berendsen
Anzeigenleitung: Philipp Schiede
Anzeigenverwaltung und Disposition: Christopher Mark (421)

So erreichen Sie die Anzeigenabteilung:
Tel. 089/46 13-494, Telefax: 089/46 13-789

Gesamtvertriebsleiter: York von Heimburg
Leiter Vertriebsmarketing: Rainer Drumm

Vertrieb Handel: MZV, Moderner Zeitschriften Vertrieb GmbH & Co. KG, Breslauer Straße 5, Postfach 11 23, 8057 Eching, Tel. 089/31 9006-0

Verkaufspreis: Das Einzelheft kostet DM 16,-

Produktion: Klaus Buck (Ltg./180), Wolfgang Meyer (Stellv./887)

Druck: SOV. Graphische Betriebe, Laubanger 23, 8600 Bamberg

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge sind in Form von Sonderdrucken erhältlich. Anfragen an Reinhard Jarczok, Tel. 089/46 13-185, Telefax 089/46 13-774

Vorstand: Otmar Weber (Vors.), Dr. Rainer Doll, Lutz Glandt

Verlagsleitung: Wolfram Höfler
Operation Manager: Michael Koeppe

Direktor Zeitschriften: Michael M. Pauly

Anschrift des Verlags: Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052, Telefax 089/46 13-100

ISSN 0931-8933

Copyright-Erklärung

Name:..

Anschrift:...

Datum:..

Computertyp:..

Benötigte Erweiterung/Peripherie:...

Datenträger: Kassette/Diskette..

Programmart:...

Ich habe das 18. Lebensjahr bereits vollendet

...., den ....

....
(Unterschrift)

Wir geben diese Erklärung für unser minderjähriges Kind als dessen gesetzliche Vertreter ab.

...., den ....

....

....

Bankverbindung:

Bank/Postgiroamt:...

Bankleitzahl:..

Konto-Nummer:...

Inhaber des Kontos:...

Das Programm/die Bauanleitung:...

das/die ich der Redaktion der Zeitschrift 64'er übersandt habe, habe ich selbst erarbeitet und nicht, auch nicht teilweise, anderen Veröffentlichungen entnommen. Das Programm/die Bauanleitung ist daher frei von Rechten anderer und liegt zur Zeit keinem anderen Verlag zur Veröffentlichung vor. Ich bin damit einverstanden, daß die Markt & Technik Verlag AG das Programm/die Bauanleitung in ihren Zeitschriften oder ihren herausgegebenen Büchern abdruckt und das Programm/die Bauanleitung vervielfältigt, wie beispielsweise durch Herstellung von Disketten, auf denen das Programm gespeichert ist, oder daß sie Geräte und Bauelemente nach der Bauanleitung herstellen läßt und vertreibt bzw. durch Dritte vertreiben läßt.

Ich erhalte, wenn die Markt & Technik Verlag AG das Programm/die Bauanleitung druckt oder sonst verwertet, ein Pauschalhonorar.

Disklader – Programme laden mit Komfort

Diskettenoberfläche de Luxe

Keine umständlichen Ladeanweisungen und ein übersichtliches Inhaltsverzeichnis der Diskette auf dem Bildschirm. Unser »Disklader« erfüllt auch gehobene Ansprüche.

von Herbert Großer

Entwicklungshelfer sind gefragt, denn noch immer sind einige Arbeitsschritte nötig, um beim C64 ein Inhaltsverzeichnis von der Diskette zu erhalten. Außerdem erschweren manche Unterdateien zu einem Programm die Übersicht im »Directory«. Genau hierfür finden Sie einen »Feuerwehrmann« auf der ersten Seite der beiliegenden Diskette – den »Disklader«. Er generiert eine Benutzeroberfläche für Ihren C64. Darin sind Funktionen integriert, wie:
- Anwahl einzelner Programme (mit jeweiliger Kurzbeschreibung),
- automatisches Laden und Starten von Diskette oder
- Erkennung der richtigen Diskette bzw. Diskettenseite.

Da sich der Disklader an erster Stelle auf der Diskette zum Sonderheft befindet, genügt es, zum Laden einzugeben:

```
LOAD":*",8
```

Nach der Bestätigung mit <RETURN> dauert es ca. 15 s, bis die Datei im Speicher ist. Sie starten mit RUN und <RETURN>. Anschließend wird das File entpackt (ca.2 s) und es erscheint die Benutzeroberfläche des »Disklader« (s. Abbildung). In der rechten unteren Bildschirmhälfte sehen Sie weiß umrandet den Namen des ausgewählten Programms. Die unterste Bildschirmzeile ist die dazugehörige Kurzerklärung. Zusätzlich finden Sie in der rechten unteren Bildschirmhälfte den Text »Seite 1 auf Disk« oder »Seite 2 auf Disk«. Da Sie die Inhaltsverzeichnisse beider Seiten (ohne die Disk zu wenden) durchblättern können, finden Sie hier



angewähltes programm — Kurzerklärung für das angewählte Programm — Seite auf der Diskette und Feld für Fehlermeldungen

den Hinweis, auf welcher Diskettenseite sich das gewählte Programm befindet.

Durch Tastendruck <CRSR aufwärts> bzw. <CRSR abwärts> wählen Sie das nächste oder vorherige Programm. Sie blättern quasi durch den Inhalt der Programme. <HOME> bringt Sie zum ersten Eintrag des Inhaltsverzeichnisses. Selbstverständlich sind nur die Programme verzeichnet, die sich eigenständig laden oder starten lassen.

<RETURN> führt Sie in den Ladeteil. Ist kein Diskettenfehler aufgetreten, erscheint kurzzeitig »00,OK, 00,00« am Bildschirm. Eventuelle Fehleranzeigen bleiben sichtbar am Bildschirm (z.B. »21,READ ERROR, 18,00«= Drive not ready). Sie lassen sich durch einen beliebigen Tastendruck wieder löschen. Schlagen Sie bitte vorher im Handbuch Ihrer Floppy nach und beseitigen Sie den Fehler. Eine andere Art der Fehlermeldung wird durch einen blinkenden Text dargestellt (z.B. »Bitte Disk wenden« oder »Falsche Diskette«). Sind Fehler ausgeblieben, lädt der Disklader das von Ihnen gewählte Programm von der Diskette und startet es. Ladefehler, die in dieser Phase auftreten, werden nicht mehr berücksichtigt: Der Disklader wird vom neuen Programm einfach überschrieben. Sonst könnten wir nur Programme veröffentlichen, die mit der Benutzeroberfläche zusammenarbeiten. Bei vielen Spielen, Tricks oder Tools ist dies aber nicht der Fall.

Für Sie bedeutet dies, nach jedem Starten eines Programms den »Disklader« erneut zu laden. Wer die Benutzeroberfläche verlassen will, gibt <RUN/STOP> ein. Sie befinden sich dann im normalen »Basic« des C64. Für einen Neustart befehlen Sie

```
SYS 12032
```

und bestätigen mit <RETURN>. Dieser Neustart funktioniert auch nach einem Reset, d.h. wenn Sie durch den entsprechenden Taster einen Hardware-Reset ausgelöst haben. Allerdings sollten Sie zwischenzeitlich kein Programm geladen haben, da dies den verwendeten Speicherbereich überschreiben könnte. Laden Sie in diesem Falle den Disklader neu.

Wir haben bei der Programmierung größten Wert auf Kompatibilität mit den unterschiedlichsten Betriebssystemerweiterungen gelegt. Lediglich bei der Gerätekonfiguration C128 mit RAM-Erweiterung und zweiter Diskettenstation sollten Sie die externe Floppy ausschalten. (gr)

Kurzinfo: Disklader

Programmart: Hilfsprogramm zum Laden der Programme auf der beiliegenden Diskette
Laden: LOAD":*",8
Starten: nach dem Laden mit RUN
Steuerung: Tastatur
Programmautor: H. Großer



Mit dem Einzelschrittsimulator analysieren Sie Maschinenprogramme und beheben eventuelle Fehler. Als besonderes Bonbon ist ein Monitor eingebaut.

Da Maschinenprogramme mit einer Geschwindigkeit von etwa 100000 Befehlen/s arbeiten, sind Programmierfehler nur sehr schwer zu analysieren. Durch den Schrittsimulator wird die Arbeitsgeschwindigkeit herabgesetzt.

Bei herkömmlichen Programmen dieser Art lassen sich die Arbeitsregister, das Stapelregister und das Programm beobachten. Doch über das Verhalten der bearbeiteten Speicherplätze lassen sie uns im unklaren. Die Fragen, ob sich das Verhalten eines Datenpuffers als richtig erweist, oder auf welchem aktuellen Stand sich ein Zeiger befindet, oder welchen Inhalt eine variable Speicherstelle ist, werden vom Programm »Schrittsimo« beantwortet.

Für den Fall, daß ein getestetes Programm Daten auf den Bildschirm ausgibt, existieren im Schrittsimo zwei Bildschirmseiten.

Standardbildschirm: Diese Seite kann, je nach Programm, in hochauflösenden Grafik- oder Textmodus geschaltet werden.

Simulatorseite: Der Simulator-Bildschirm ist in vier Bereiche unterteilt (Abb. 1).

1. Im Prozessorteil werden die Arbeitsregister in hexadezimaler und das Statusregister in binärer beziehungsweise hexadezimaler Form angezeigt (Zeile 3 und 4 in Abb. 1). Die obersten 13 Elemente des Stapelregisters sind in den Zeilen 6 und 7 mit den zugehörigen Zählerständen dargestellt.

2. Das Programm erlaubt das Beobachten des aktuellen Befehls (Zeile 9), des nächsten (Zeile 10) und des darauf folgenden Befehls (Zeile 11).

3. Durch die Bereiche können jeweils zehn aufeinanderfolgende Speicherabschnitte im ASCII- oder Bildschirmcode verfolgt werden (nach Angabe der ersten Adresse).

4. Bytes lassen sich in hexadezimaler, dezimaler und dualer Form und im ASCII-Format kontrollieren.

Um die Fehlersuche zu beschleunigen, können fehlerfreie Unterprogramme in Originalgeschwindigkeit ausgeführt werden.

Nach dem Start des Simulators wird zusätzlich noch ein Monitor ins Betriebssystem integriert. Dieser Monitor unterscheidet sich, was die Handhabung betrifft, von herkömmlichen Monitorprogrammen. Beispielsweise werden alle Monitorbefehle mit einem Punkt am Anfang der Zeile gekennzeichnet. Dadurch ist der Umschaltvorgang (Monitor/Basic) nicht erforderlich. Die Monitor-Befehle lassen sich aus Basic aufrufen und werden sofort dargestellt.

Da der Bildschirm in Arbeitsbereiche unterteilt ist, wird ein Einfügen beziehungsweise Löschen von Befehlen wesentlich vereinfacht. Es können 16 Befehlsbereiche und zu jedem 16 Zeichen (Text) eingeben werden. Der momentan gültige wird in der ersten Bildschirmzeile dargestellt (Abb. 2). Arbeitet man außerhalb definierter Bereiche, erscheint die Meldung »out of memory«. In einem solchen Fall sind Verschiebungen nicht möglich.

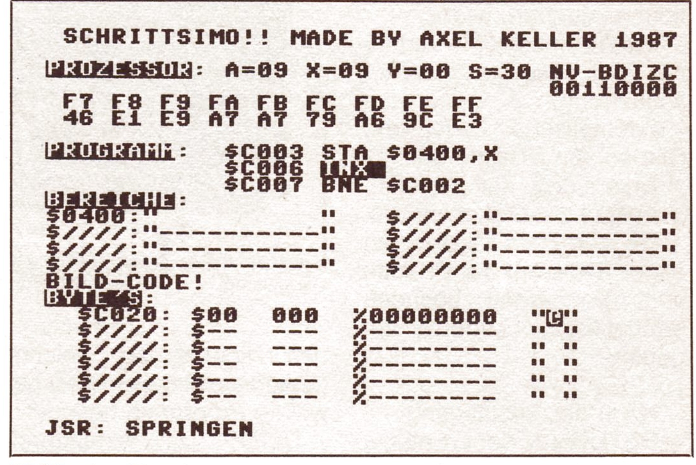
Durch den Befehl ».M« wird der Monitor auf Memory-Betrieb umgeschaltet, das heißt Speicherbereiche werden in hexadezimaler Form dargestellt (Abb. 3).

Der Befehl ».A« aktiviert den Assembler-Betrieb wieder. Danach sind wieder Eingaben für Assemblerbefehle und die disassemblierten Ausgaben möglich.

Durch einen einfachen Speicherbefehl »S$startdresse$-endadresse "name",geräteadresse« läßt sich ein Programm auf Diskette oder Kassette speichern. Beim Laden solcher Programme ist man nicht auf den Monitor angewiesen.

».G adresse« startet ein zuvor geladenes oder eingegebenes Maschinenprogramm. »adresse« darf dabei in hexadezimaler (vorangestelltes $) oder dezimaler Schreibweise verwendet werden.

Bei der Druckerausgabe werden Branch-Befehle mit Richtungspfeilen versehen (Abb. 4). Dies ermöglicht eine übersichtliche Darstellung von Programmen und erleichtert die Fehlersuche. Fehladressierungen werden vom Programm erkannt und durch einen Stern signalisiert.



[1] Mit der Simulatormaske lassen sich neben den Registern auch die Inhalte frei definierbarer Speicherbereiche und einzelner Bytes beobachten

```
.B3:$8800-$88DA "BEFEHL IN ASCII"
.$8800
.$8800 LDA #$40
.$8802 STA $22
.$8804 LDA #$03
.$8806 STA $23
.B
.B0:$89D5-$89FF "TEXT BER. SCHRE."
.B1:$8A49-$8ADA "BEFEHL 'B'"
.B2:$8D5A-$8BAE "BEFEHL 'E'"
.B3:$8800-$88DA "BEFEHL IN ASCII"
.B4:$8BB0-$8C1E "BEFEHL 'A'"
.B5:$93F0-$93FF "SPALTE LOESCHEN"
.B6:$9400-$947A "HINTERGRUND AUFB"
.B7:$9650-$9674 "BYTE AUSGEBEN"
.B8:$96A0-$96C0 "X-AKTUALISIEREN"
.B9:$7000-$8000 "VERSUCH!!!"
.BA:$0000-$0000 ""
.BB:$0000-$0000 ""
.BC:$0000-$0000 ""
.BD:$0000-$0000 ""
.BE:$0000-$0000 ""
.BF:$0000-$0000 ""
.$8806 STA $23
```

[2] Das Arbeiten mit Bereichen erhöht die Übersichtlichkeit. Der aktuelle Bereich wird in der obersten Zeile gezeigt.

Bedienung

Initialisieren und Parameter setzen: Das Programm wird mit

```
LOAD "SCHRITTSIMO",8,1
```

geladen. Danach zunächst NEW <RETURN> einzugeben, um die Basic-Zeiger zu setzen. Initiiert wird mit SYS40000. Dann erscheint zuerst die Arbeitsmaske des Simulators mit willkürlichen Einträgen. Mit <F4> wird sie mit Standardparametern besetzt und mit <F3> wieder verlassen. Die weiteren Funktionen erfahren Sie unter »Starten des Simulators«.

Nach einem Systemreset ist der Initialisierungsvorgang zu wiederholen.

Der Monitor: Die Arbeitsweise des Assemblers und Disassemblers läßt sich am besten anhand eines Programms demonstrieren:

```
1. Befehl: .$C000 LDX #$0
2. Befehl: .$C002 TXA
3. Befehl: .$C003 STA $400,X
4. Befehl: .$C006 INX
5. Befehl: .$C007 BNE $C002
6. Befehl: .$C008 RTS
```

Eingabe eines Befehls: Monitor-Befehle beginnen im Unterschied zu Basic-Befehlen mit einem Punkt. Der erste Assembler-Befehl (oben) wird mit folgender Syntax eingegeben:

```
.$C000 ldx #$0
```

Dem Punkt folgt unmittelbar ein »$«-Zeichen (weist auf eine hexadezimale Zahl hin) und die hex-Zahl mit der Befehlsadresse. Nach einer Leerstelle steht der eigentliche Assembler-Befehl. Dieser wird unmittelbar nach <RETURN> in einen Maschinencode übersetzt und ab der angegebenen Adresse gespeichert. Danach erscheint eine Zeile tiefer die nächste Adresse mit dem dort vorhandenen Assembler-Code, bzw. das dort gespeicherte Byte. Den zweiten Befehl des Beispielprogramms schreiben Sie hinter die gerade ausgegebene Adresse - Sie überschreiben damit einfach vorgegebenen Befehl und übernehmen die neue Eingabe mit <RETURN>.

Bei Sprungbefehlen ist das Ziel als hex-Adresse anzugeben. Bei Fehleingaben erhalten Sie die Meldung »syntax error?«.

Disassemblieren eines Programms: Will man einen Speicherbereich disassemblieren, ist die gewünschte Adresse wie folgt anzugeben:

```
.$c000
```

Nach der Übernahme mit <RETURN> erscheint der dazugehörige Befehl auf dem Bildschirm. Da alle Tasten auf Wiederholen gestellt sind, kann durch ständiges Halten der RETURN-Taste das ganze Programm disassembliert werden.

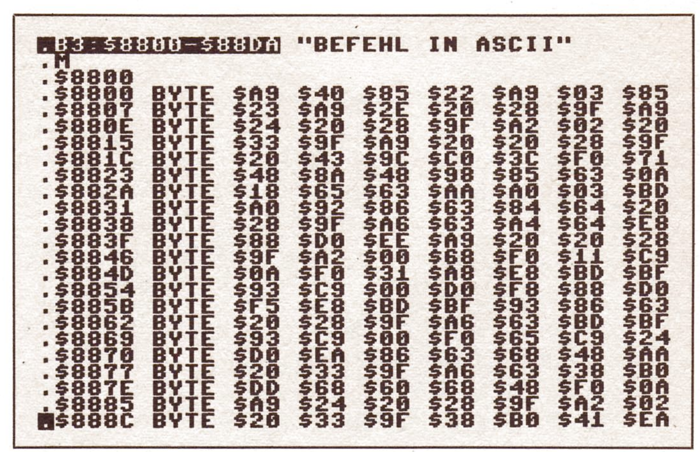
Arbeitsbereiche

Zum Einfügen oder Löschen eines Befehls ist vorher der Bereich festzulegen. Es lassen sich max. 16 dieser Bereiche definieren und jedem ein Text von 16 Zeichen zuordnen. Dadurch läßt sich ein Programm übersichtlich aufgliedern.

Definition eines Bereichs:

```
.bx:$yyyy-$zzzz "16 zeichen text"
B = Bereich
x = Bereichsnummer
y = Bereichsanfang
z = Bereichsende
```

Arbeitet man in diesem Bereich, wird er in der obersten Bildschirmzeile angezeigt. Befindet man sich außerhalb definierter Bereiche, erscheint in dieser Zeile die Meldung »out of



```
.B3:$8800-$88DA "BEFEHL IN ASCII"
.M
.$8800
.$8800 BYTE $A9 $40 $85 $22 $A9 $03 $85
.$8807 BYTE $23 $A9 $2E $20 $28 $9F $A9
.$880E BYTE $24 $20 $28 $9F $A2 $02 $20
.$8815 BYTE $33 $9F $A9 $20 $20 $28 $9F
.$881C BYTE $20 $43 $9C $C0 $3C $F0 $71
.$8823 BYTE $48 $8A $48 $98 $85 $63 $0A
.$882A BYTE $18 $65 $63 $AA $A0 $03 $BD
.$8831 BYTE $A0 $92 $86 $63 $84 $64 $20
.$8838 BYTE $28 $9F $A6 $63 $A4 $64 $E8
.$883F BYTE $88 $D0 $EE $A9 $20 $20 $28
.$8846 BYTE $9F $A2 $00 $68 $F0 $11 $C9
.$884D BYTE $0A $F0 $31 $A8 $E8 $BD $BF
.$8854 BYTE $93 $C9 $00 $D0 $F8 $88 $D0
.$885B BYTE $F5 $E8 $BD $BF $93 $86 $63
.$8862 BYTE $20 $28 $9F $A6 $63 $BD $BF
.$8869 BYTE $93 $C9 $00 $F0 $65 $C9 $24
.$8870 BYTE $D0 $EA $86 $63 $68 $48 $AA
.$8877 BYTE $20 $33 $9F $A6 $63 $38 $B0
.$887E BYTE $DD $68 $60 $68 $48 $F0 $0A
.$8885 BYTE $A9 $24 $20 $28 $9F $A2 $02
.$888C BYTE $20 $33 $9F $38 $B0 $41 $EA
```

[3] Durch den Befehl ».M« werden alle Speicherinhalte in hexadezimaler Form ausgegeben

memory«. Einzelne Bereiche dürfen sich nicht überschneiden. Tritt eine Überschneidung auf, wird der eingeschnittene Bereich eine Zeile tiefer angezeigt. Im Beispiel:

```
.b0:$c000-$c01f "beispielprogramm"
```

Bereiche aufrufen:

```
.b    listet alle bereiche (von 0 bis 15)
.bx   listet den bereich »x«
.b-x  listet die bereiche 0 bis »x«
.bx-  listet die bereiche »x« bis 15
.bx-y listet die bereiche »x« bis »y«
```

Bereiche löschen: So, wie sich Bereiche aufrufen lassen, werden sie auch gelöscht. Die Befehlssyntax dafür lautet:

.lb; .lbx; .lb-x; .lbx-; .lbx-y.

Einfügen und Löschen

Löschen: Um den dritten Befehl in unserem Beispiel zu löschen, sind die Adressen und die Anzahl der Befehls-Bytes anzugeben:

```
.$c003 a$3
```

Nach <RETURN> werden alle nachfolgenden Bytes im definierten Bereich um minus drei Stellen verschoben. In den freiwerdenden Speicherstellen steht nach dem Löschen der Wert $FF.

Die Anzahl der zu verschiebenden Bytes läßt sich bis auf $ff erhöhen. Dadurch können mehrere Befehle auf einmal gelöscht werden. Wenn sich die Adresse außerhalb der Bereiche befindet oder die Anzahl der Bytes die zu verschiebenden Speicherstellen (Bytes bis Bereichsende) übersteigt, wird die Fehlermeldung »out of memory error?« ausgegeben.

Einfügen: Jetzt soll der Befehl LDX $400,X an seiner ursprünglichen Stelle eingefügt werden. Hierzu sind die Adresse und die Byte-Größe des Befehls anzugeben:

```
.$c003 e$3
```

Danach werden alle Bytes ab der angegebenen Adresse im entsprechenden Bereich um drei Stellen vorgeschoben.

Anmerkung: Die Anzahl der eingefügten Bytes wird am Ende des Bereichs gelöscht. Anschließend geben Sie ein:

```
.$c003 ldx $400,x
```

Memory- und Assembler-Betrieb: Nach dem Umschalten durch ».M« auf Memory-Betrieb werden alle Speicherinhalte in hexadezimaler Form ausgegeben. Durch den Befehl ».A« läßt sich der ursprüngliche Assemblierzustand wiederherstellen. Diese Funktion steuert nur die Ausgabe, das heißt es kann in jedem Zustand Hexadezimal- oder Assemblercode eingegeben werden. Die Anfangswerte für die Ausgabe setzen Sie durch Eingabe des hex-Werte. Beispielsweise beginnt

```
.$c000
```

mit der Ausgabe (».M« oder ».A«) ab dieser Speicherstelle.

```
.B9:$7000-$7020 "VERSUCH!!!'

.$7000 LDY #$00
.$7002 LDX #$04
.$7004 STY $FA
.$7006 STX $FB
.$7008 TYA                <---------┐
.$7009 STA ($FA),Y                | |
.$700B INY                        | |
.$700C BNE $7008          >-------┘ |
.$700E INC $FB                      |
.$7010 DEX                          |
.$7011 BNE $7008          >---------┘
.$7013 RTS
```

[4] Bei der Druckerausgabe werden Branch-Befehle mit Richtungspfeilen versehen

Speicherfunktionen: Ein Maschinenprogramm läßt sich nach Angabe der Anfangs- und Endadresse speichern: Speichern auf Datasette mit: .S$xxxx-$yyyy ″16 Zeichen Text″

Speichern auf Diskette mit: .S$xxxx-$yyyy ″16 Zeichen Text″,8

```
S = Save
x = Anfangsadresse
y = Endadresse
```

Das Laden erfolgt mit dem normalen Basic-Befehl LOAD- ″name″,8,1 beziehungsweise für Datasette LOAD ″name″1,1

Ausdrucken eines Programms: ».Dx« startet die Druckerausgabe. »x« kennzeichnet den auszugebenden Bereich. Die Branchebefehle werden mit Richtungspfeilen versehen. Bei eventueller Fehladressierung erscheint ein Stern. Der Befehl läßt sich wie folgt variieren:
.d; .dx; .dx-; .d-x; .dx-z

Starten eines Programms: Mit ».g$xxxx« läßt sich ein Maschinenprogramm starten. »xxxx« kennzeichnet die Startadresse in hexadezimaler Form. Beim Starten des Beispielprogramms mit ».G$C000« wird in Bruchteilen einer Sekunde der komplette Zeichensatz ausgegeben.

Simulator

Testen wir als nächstes das Beispielprogramm Schritt für Schritt. Um die Effektivität des Simulators besser erläutern zu können, ist das Programm wie folgt zu ergänzen:

```
1. Befehl: .$C000 LDX #$0
2. Befehl: .$C002 TXA
3. Befehl: .$C003 STA $400,X
4. Befehl: .$C006 INX
5. Befehl: .$C007 BNE $C002
6. Befehl: .$C009 LDA #$0
7. Befehl: .$C00B STA $C020
8. Befehl: .$C00E DEC $C011
9. Befehl: .$C020 RTS
```

Starten des Simulators: Wird der Einzelschrittsimulator mit SYS40000 gestartet, erscheint sofort die Arbeitsmaske auf dem Bildschirm.

<F7> – Parameter festlegen:

... führt in den Eingabemodus. Im Eingabefeld für das Register A erscheint der Cursor. Jetzt kann der gewünschte hexadezimale Wert mit den Tasten 0 bis 9 und A bis F eingegeben oder mit <CRSR auf/nieder> eingestellt werden. Mit <CRSR rechts/links> läßt sich der Cursor im Eingabefeld positionieren. <RETURN> führt ins nächste Eingabefeld. Im Stapelbereich kann durch <RETURN> nur der oberste Wert des Stapels erreicht werden. Möchte man die unteren Stapelelemente ändern, ist dies nur mit <CRSR rechts> möglich.

Bereich- und Byte-Sektor: Diese beiden Sektoren dienen zur Beobachtung von Speicherinhalten während des Programmtests. In den acht zur Verfügung stehenden Bereichen werden jeweils 10 Byte im ASCII- oder Bildschirmmodus dargestellt. Im Byte-Sektor können die Inhalte sechs verschiedener Speicherzellen während eines Tests in hexadezimaler, dezimaler und binärer Form kontrolliert werden. Zusätzlich erscheint noch das zum Byte gehörende ASCII- beziehungsweise Bildschirmzeichen. Soll ein Feld im Bereich- oder Byte-Sektor gesetzt werden, reicht es, den Cursor mit Hilfe der RETURN-Taste auf das erste »/«-Zeichen hinter dem »$«-Zeichen zu positionieren, und wenn man die gewünschte Adresse eingegeben hat, »/«-Taste zu drücken. Um das Beispielprogramm zu testen, müssen jetzt die Parameter eingerichtet werden. Dazu ist der Zähler im »Programm«-Teil auf die Anfangsadresse des zu testenden Programms zu setzen; in unserem Fall also auf $C000. Danach stellt man das erste Bereichsfeld auf $C0400. Diese Adresse entspricht der linken, oberen Ecke der Standardbildschirmseite. Das erste Byte-Feld muß noch auf $C020 gesetzt werden, um den Inhalt der Speicherzelle zu kontrollieren. Nun ist so oft die RETURN-Taste zu drücken, bis der Cursor verschwindet.

<F6> – ASCII- und Bildschirmcode: Der Bildschirmcode entspricht den Zeichen auf dem Bildschirm, die durch direktes Einlesen einer Zahl in das Video-RAM auf dem Bildschirm erscheint.

Beim ASCII-Code werden Zahlen über die genormte ASCII-Tabelle in den Bildschirmcode gewandelt.

Durch das Umschalten mit <F6> kann der Bereichssektor im ASCII- oder Bildschirmcode beobachtet werden.

<F8> – Testvorgang: Im »Programm«-Teil steht jetzt invertiert der erste Befehl des Beispielprogramms. Den zweiten Befehl findet man eine Zeile tiefer. Wenn der Bereichssektor auf Bildschirmcode geschaltet ist, sind die ersten zehn Zeichen der Standardbildschirmseite sichtbar. Der Inhalt der Speicherzelle $C020 ist bis jetzt noch nicht definiert.

Mit <F8> wird der invertierte Befehl durch den simulierten Prozessor bearbeitet. Das X-Register ist jetzt in jedem Fall $00, und das Zeroflag ist im Statusregister gesetzt. Zugleich wird der erste Befehl eine Stelle nach oben geschoben (hier ist immer der zuletzt bearbeitete Befehl zu sehen) und der zweite Befehl wandert in die Mitte, gefolgt vom dritten Befehl. Im Bereichs- oder Byte-Sektor ist noch keine Änderung zu registrieren. Beim erneuten Drücken von <F8> wird das A-Register auf $00 gesetzt. Erst beim Bearbeiten des dritten Befehls erkennt man im ersten Feld des Bereichs das Zeichen »@«. Der vierte Befehl erhöht das X-Register um 1. Somit wird auch das Zeroflag des Statusregisters gelöscht. Der nächste Sprungbefehl wird durchgeführt, wenn das Zeroflag gelöscht ist. Nach Ausführung erscheint der Sprungbefehl in der oberen Zeile. Die invertierte Zeile weist dann das Sprungziel (zweiter Befehl) auf und darunter ist wieder der dritte Befehl zu erkennen.

Drückt man ständig <F8>, wird das Programm schneller bearbeitet. Um ein Programm zu testen, muß der Computer vor jedem Testbefehl auf den Ursprungzustand eingestellt werden. Das heißt, das Programm schaltet bei jedem Befehl auf den Standardbildschirm und aktiviert danach wieder die Arbeitsmaske.

<F1> – Simulatormaske und Standardbildschirm: Ausgaben auf den Standardbildschirm können durch Umschalten auf die Standardbildschirmseite beobachtet werden. Schaltet ein zu testendes Programm auf eine andere Bildschirmseite oder auf hochauflösende Grafik um, wird auch beim Umschalten mit <F1> auf diese Seite geschaltet. Selbst im umgeschalteten Zustand werden durch <F8> Testbefehle bearbeitet.

<F5> – JSR testen oder ausführen: Führt eine Verzweigung in ein Unterprogramm durch. Dabei wird die aktuelle Adresse auf den Stapel gelegt und der Programmzähler mit der Zieladresse geladen. Dies entspricht dem Verhalten des Prozessors bei einer Unterprogrammverzweigung.

Steht bereits fest, daß das Unterprogramm fehlerfrei ist, kann es in Originalgeschwindigkeit bearbeitet werden. Ob man mit <F5> auf »springen« oder »ausführen« geschaltet hat, ist in der untersten Zeile zu erkennen. Diese Anzeige ist maßgebend für die Art der Bearbeitung.

<F4> – Reset: Beim Drücken von <F4> werden verschiedene Parameter zurückgesetzt:
- Das Umschalten mit <F1> wird auf die Standardbildschirmseite (Einschaltzustand) gelenkt.
- Der simulierte Stapel entspricht dem aktuellen Originalstapel.

<F2> – Bild-Neuaufbau:
... löscht die Arbeitsmaske und baut sie wieder neu auf. Dabei werden die alten Werte übernommen.

<F3> – Ende des Testvorgangs:
... führt zum Basic-Interpreter. Im Gegensatz zu Basic V2.0 sind die Monitorbefehle aktiv.

Tips

Der Einzelschrittsimulator funktioniert nicht bei Maschinenprogrammen mit gesetztem Interruptflag (SEI).

Da Zugriffe auf die Zeropage stattfinden, müssen die intern benutzten Speicherzellen in andere Bereiche gesichert werden. Vor jedem Testbefehl wird also die Zeropage mit den gesicherten Werten geladen. Sie sollten daher bei folgenden Zeropagewerten auf den Sicherungsbereich zugreifen:

```
$FA bis $FB = $84A1 bis $84A2
$7A bis $7B = $84A3 bis $84A4
$22 bis $23 = $84A5 bis $84A6
$61 bis $69 = $8431 bis $8439
```

Wenn der Drucker falsche Zeichen ausgibt, können sie an folgenden Speicherstellen geändert werden:

```
"|"  senkrechter Balken :  $8F02
"—"  waagerechter Balken:  $8E8C
">"  Pfeil für Start:      $8EA4
"<"  Pfeil für Ziel:       $8EAF
"┴"  Auf-Verzweigung:      $8EBE
"┬"  Ab-Verzweigung:       $8EB4; $8ECE
"┘"  Ecke nach oben:       $8ED2
"┐"  Ecke nach unten:      $8ECC
"*"  Zeichen für Fehler:   $8F26
"+"  Kreuzung:             $8F08
```

Außerdem lassen sich die Bildschirmfarben frei wählen: Standardseite: Hintergrund = $9271; Rahmen = $9270; Schrift = $9FC8.

Simulatorseite: Hintergrund = $9241; Rahmen = $9240

Nun viel Spaß mit dem Programm. (gr)

Kurzinfo: Einzelschrittsimulator

Programmart: Test-Tool für Maschinenprogramme
Laden: LOAD"SCHRITTSIMO",8,1
Initiieren: nach dem Laden SYS40000 und <RETURN> eingeben
Besonderheiten: eingebauter Monitor
Benötigte Blocks: 25
Programmautor: Axel Keller

Decompiler

Zurück zum Original

Compilierte Basic-Programme lassen sich nachträglich nicht mehr ändern. Dazu muß das Compilat wieder zum Basic gewandelt werden – mit dem Decompiler.

So etwas kann jedem passieren: Ein tolles Basic-Programm ist entstanden und wurde mit dem Austro-Compiler schneller gemacht. Wochen später wollen Sie etwas ändern und stellen fest, daß das Basic nicht mehr aufzutreiben ist. Änderungen sind also nicht mehr möglich. Was tun? Sie nehmen den Decompiler und generieren aus dem Compilat wieder den Originaltext.

Geladen wird das Programm mit

```
LOAD"DECOMPILER",8
```

und gestartet mit RUN. Danach erscheinen die Meldungen »Decompiler zum Austrocompiler« und »Bitte Filenamen eingeben« am Bildschirm. Jetzt sollten Sie eine Arbeitsdiskette einlegen, auf der sich das zu bearbeitende Programm befindet. Tippen Sie anschließend den File-Namen ein und bestätigen Sie mit <RETURN>. Achtung: Der File-Namen darf nicht mehr als 13 Zeichen lang sein.

Das Programm erkennt, ob es sich um ein Compilat vom Austro-Compiler handelt. Achtung: Es gibt mehrere Versionen des Austro-Compilers, wovon jedoch nur eine bearbeitet wird. Sollten Sie die falsche Version besitzen, erscheint die gleiche Fehlermeldung wie bei einem falschen Compiler.

Funktionsweise

Der Decompiler geht in drei Arbeitsschritten vor:

Pass 1 – Arrays, GOTO, GOSUB, IF
... legt eine Tabelle der Befehle mit den entsprechenden Zeilennummern an. Hier werden auch die Namen der Variablen und Arrays gesucht und notiert.

Pass 2 – Zeilenanalyse
... analysiert die Struktur des Basic-Programms und legt eine Cross-Referenz-Liste an. Dabei werden überlange Zeilen (mit mehr als 80 Zeichen) aufgeteilt.

Pass 3: – Basic-Programm
... erzeugt aus allen Angaben von Pass 1 und Pass 2 das Basic-Programm und numeriert es in Zehnerschritten durch.

Das generierte Basic befindet sich auf der Arbeitsdiskette unter dem Originalnamen mit vorangestellten »B/«. Haben Sie als Programmnamen »TEST« eingegeben, heißt das recompilierte Programm jetzt »B/TEST«.

Ist der Decompiler mit dem Übersetzen fertig, fragt er nach dem nächsten Programmnamen. Wenn Sie noch weitere Rückwandlungen vornehmen wollen, ist das hier im nächsten Arbeitsgang möglich. Mit <PFEILLINKS> beenden Sie. (gr)

Kurzinfo: Decompiler

Programmart: Recompiler zum Austro-Compiler
Laden: LOAD"DECOMPILER",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: erkennt den Austro-Compiler
Benötigte Blocks: 26
Programmautor: Thomas Kläger

Komfort großgeschrieben

»ExtEditor« bietet Ihnen eine professionelle Programmierumgebung und beim Start sogar ein Installationsmenü – da kommen PC-Gefühle auf.

Ein Vorteil des C64 ist der eingebaute Basic-Interpreter, der sofort nach dem Einschalten Gewehr bei Fuß steht. Allerdings läßt sein Komfort zu wünschen übrig. Es fehlen etliche Befehle, beispielsweise die Anzeige des Floppyinhaltsverzeichnisses am Bildschirm. Außerdem wurde nicht allzuviel Wert auf die Eingaberoutinen zur Basic-Programmierung und im Direktmodus gelegt. Diese Mißstände lassen sich umgehen, wenn Sie mit

```
LOAD"EXTEDITOR",8
```

von der beiliegenden Diskette laden und mit RUN starten. Die Betriebssystemerweiterung meldet sich nach einem kurzen Intro mit einem Auswahlmenü. Darin lassen sich ein Installationsmenü (»Install Menu <SPACE>«) und ein komfortabler Editor (»Start Editor <RETURN>«) über die angezeigten Tasten anwählen.

Installationsmenü (Abb.)

...dient zur Grundeinstellung der Bildschirmfarben, Funktionstastenbelegung, Editier-Windows, Peripheriegeräte. Zusätzlich läßt sich TurboTape (beschleunigte Kassettenroutinen) mit aktivieren. Die aktive Option ist revers gekennzeichnet und läßt sich mit <F1> und <F7> auswählen. <F3> und <F5> ändern die Parameter.

Colors - Optionen zur Farbauswahl

...enthält einen Farbbalken zur Überprüfung der eingestellten Farben. Die Pfeile neben den einzelnen Farboptionen werden über <F2> und <F3> gewählt, die Optionen mit <F1> und <F7>.

border - Farbe des Bildschirmrahmens
background - Hintergrundfarbe
window - Farbe des Fensters
character - Schriftfarbe

Window - Fensteroptionen

...dient zur Positionierung des Eingabe-Windows.

up - Zeilenanzahl des Fensters
down - Position am Bildschirm (von oben)

Funktionskeys - Funktionstastenbelegung

...legt den Befehl fest, der per Tastendruck am Bildschirm ausgegeben wird. Die Befehle entsprechen den neuen von ExtEditor und lassen sich mit <F3> und <F7> auswählen.

Belegen lassen sich die darunter angezeigten Tasten F2, F4, F6 und F8.

Default Device - I/O-Gerät

...wählt die Gerätenummer des für Lade-Speicher-Funktionen zuständigen Geräts:

0 = Standard
1 = TurboTape
8 = TurboDisk

Printer Device - Druckernummer

...wählt die Gerätenummer für den Drucker. Optional sind »4« oder »5«.

Install TurboTape - Routinen ein/aus

...installiert bei »No« keine TurboTape-Routinen. In diesem Fall steht ein halbes KByte mehr Basic-Speicher zur Verfügung.

Wenn sie Ihre persönlichen Grundwerte eingestellt haben, drücken Sie <RETURN>. Anschließend folgt die Frage:

»Save Configuration (Y/N) ?«. Bevor Sie diese mit <Y> beantworten, sollten Sie sicherstellen, daß Ihre Diskette keinen Schreibschutz besitzt (Sicherheitskopie verwenden, beiliegende Diskette ist gegen Beschreiben geschützt!). Danach werden die geänderten Grundwerte auf Diskette gesichert.

Der komplette Speichervorgang wird bei »N« übersprungen. Anschließend initialisiert sich das Programm mit den von Ihnen eingestellten Parametern.

Zwei Modi

Der jetzt betriebsbereite Editor besitzt zwei Funktionsblöcke, zwischen denen Sie mit <CONTROL> umschalten können: Editier- und Command-Modus. In jedem bietet sich ein eigenes Fenster an. Für das Command-Window sind die untersten beiden Bildschirmzeilen vorgesehen, das Editierfenster läßt sich in Größe und Lage bestimmen. Anhand der Farbe (Window-Farbe) sehen Sie, in welchem Fenster Sie sich befinden.

Editiermodus

...dient zur Eingabe und Änderung von Basic-Zeilen. Dabei wurden die Editierfähigkeiten des Basic-Interpreters erheblich erweitert. So läßt sich das Listing beispielsweise aufwärts und abwärts scrollen und seitenweise vorwärts und rückwärts blättern:

<**CRSR aufwärts**>

...an der obersten Window-Zeile wird der Bildschirminhalt nach unten gescrollt und entsprechend die vorherige Basic-Zeile ausgegeben.

<**CRSR abwärts**>

...an der untersten Window-Zeile wird der Bildschirminhalt nach oben gescrollt und die nächste Basic-Zeile ausgegeben.

<**CLR/HOME**>

...bringt den Cursor an die linke obere Ecke des Windows.

<**SHIFT CLR/HOME**>

...wie <CLR/HOME>, zusätzlich wird der Bildschirm gelöscht.

<**CTRL Z**>

...bringt den Cursor an die linke untere Ecke des Windows.

<**CTRL Pfeillinks**>

...löscht das Zeichen unter dem Cursor (wie DEL beim PC).

<**CTRL L**>

...fügt eine Leerzeile ein.

<**CTRL D**>

...löscht eine Zeile. Bei einer Basic-Zeile wird diese auch aus dem Programm gelöscht.

<**CTRL I**>

...fügt Zeichen in den Text ein, ohne sie zu überschreiben, schiebt also die bestehenden Zeichen nach links. Die Rahmenfarbe wird während dieser Option wie die Hintergrundfar-

be geschaltet. Ein erneutes <CTRL I> schaltet zurück auf den normalen Schreibmodus.

<**CTRL W**>

...löscht das Wort unter dem Cursor.

<**CTRL A**>

...löscht die Zeile ab einschließlich Cursor-Position (nach links).

<**CTRL B**>

...löscht die Zeile ab Cursor-Position ohne das Zeichen unter dem Cursor.

<**CTRL P**>

...druckt die Bildschirmseite auf dem Gerät, das unter PDEV deklariert ist (Printer Device, s. »Installationsmenü« und »Befehle«).

<**F1**>

...listen der ersten Programmseite.

<**F3**>

...listen der Programmseite vor der gerade sichtbaren.

<**F5**>

...listen der Programmseite nach der gerade sichtbaren.

<**F7**>

...listen der letzten Programmseite.

<**F2**>, <**F4**>, <**F6**> und <**F8**>

...führt die im Installationsmenü (s. d.) definierten Kommandos aus und schaltet in den Kommandomodus.

Command-Modus

...dient zur Eingabe von Direktbefehlen. Hierzu stehen die direkt ausführbaren Befehle des Basic V 2.0 zur Verfügung (s. »Handbuch C 64«) und zusätzlich 37 neue ExtEditor-Befehle (s. »SHOW«).

Wenn Sie nur die Zeilennummer eingeben, erscheint eine Bildschirmseite des Listings, beginnend mit dieser Nummer.

<RETURN> ohne andere Eingaben wiederholt den letzten Befehl (kann anschließend mit <RETURN> übernommen werden).

Die Bildschirmausgabe läßt sich auf einen angeschlossenen Drucker umleiten, wenn Sie vor den Befehl ein »!« stellen. So gibt

```
!list 100-200
```

die Basiczeilen 100 inkl. 200 auf einem Printer aus. Dieser Druckmodus funktioniert natürlich nicht bei Diskettenbefehlen.

16 Tastenfunktionen erleichtern die Befehlseingabe:

<**CRSR aufwärts**>

...stoppt an der obersten Window-Zeile.

<**CRSR abwärts**>

...stoppt an der untersten Window-Zeile.

<**CLR/HOME**>

...bringt den Cursor an die linke obere Ecke des Windows.

<**SHIFT CLR/HOME**>

...wie <CLR/HOME>, zusätzlich wird das Command-Window gelöscht.

<**CTRL Pfeillinks**>

...löscht das Zeichen unter dem Cursor (wie DEL beim PC).

<**CTRL L**>

...fügt eine Leerzeile ein.

<**CTRL D**>

...löscht eine Zeile

<**CTRL I**>

...fügt Zeichen in den Text ein ohne sie zu überschreiben, schiebt also die bestehenden Zeichen nach links. Die Rahmenfarbe wird während dieser Option wie die Hintergrundfarbe geschaltet. Ein nochmaliges <CTRL I> schaltet zurück auf den normalen Schreibmodus.

<**CTRL W**>

...löscht das Wort unter dem Cursor.

<**CTRL A**>

...löscht die Zeile ab einschließlich Cursor-Position.

<**CTRL B**>

...löscht die Zeile ab Cursor-Position ohne das Zeichen unter dem Cursor.

<**CTRL P**>

...druckt die Bildschirmseite auf dem Gerät, das unter PDEV deklariert ist (Printer Device, s. »Installationsmenü« und »Befehle«).

<**F1**>

...listen der ersten Programmseite.

<**F3**>

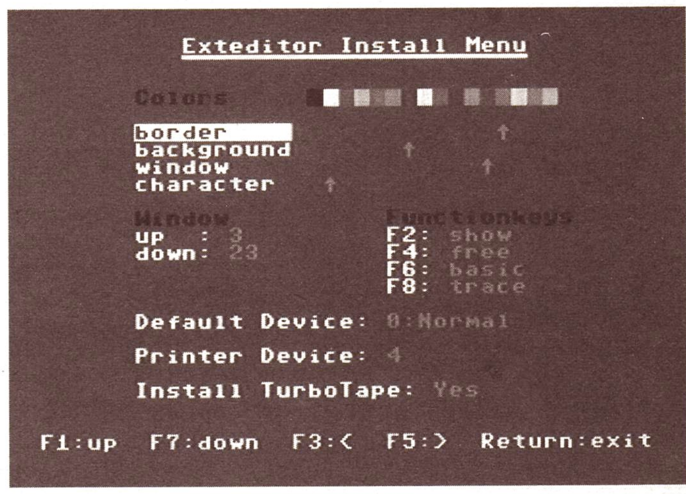
...listen der Programmseite vor der gerade sichtbaren.

<**F5**>

...listen der Programmseite nach der gerade sichtbaren.

<**F7**>

...listen der letzten Programmseite.



Die eingestellten Standardparameter werden bei jedem neuen Start des ExtEditor übernommen

Neue Befehle

ExtEditor stellt 37 neue Befehle zur Verfügung. Um größtmögliche Kompatibilität mit Basic V 2.0 zu erreichen, sind diese nur aus dem Command-Modus erreichbar (Direkt-Befehle). Die Kommandos werden ohne <SHIFT> eingegeben:

Grundparameter

Mit diesen Befehlen lassen sich die Standardparameter des Installationsmenüs ändern.

DEV Kennzahl

...bestimmt das Standard-I/O-Gerät:

0 = Standard
1 = Turbotape
8 = TurboDisk

```
PDEV8
```

setzt die Standardparameter auf TurboDisk

COLOR Rahmen,Hintergrund,Window,Schrift

...bestimmt die Grundfarben des ExtEditor.

```
COLOR0,1,2,6
```

setzt Rahmenfarbe auf Schwarz (0), Hintergrundfarbe auf Weiß (1), Window-Farbe auf Rot (2) und Schriftfarbe auf Blau (6). Achtung: Die Schriftfarbe sollte einen guten Kontrast zur Window- und Hintergrundfarbe bieten.

WINDOW up,down

...definiert das Editorfenster. Dabei gibt »up« die Zeile oben (ab der das Window beginnt) an, »down« die Zeile unten:

```
WINDOW1,12
```

ergibt ein Editorfenster, das von Zeile 1 bis Zeile 12 reicht.

PDEV

...setzt die Gerätenummer des Druckers (4 oder 5). Genützt wird diese Option von <CTRL P> und »!«.

Allgemeine Befehle

SHOW
...listet alle ExtEditor-Befehle

BASIC
...listet alle Basic-Befehle

FREE
...gibt eine Übersicht über die vom Programm belegten (bytes used), noch freien (bytes free) und insgesamt verfügbaren Basic-Bytes (bytes available).

MERGE dateiname,device
...lädt ein Programm in das im Speicher befindliche.

APPEND
...fügt ein Programm hinter das im Speicher befindliche

EXIT
...verläßt ExtEditor und führt einen RESET durch.

Programmierhilfen

RENUM Neustart,Schritt,von-bis
...numeriert ein Basic-Programm im Speicher neu. Dabei wird in »Neustart« die Nummer der ersten Zeile, mit »Schritt« die Schrittweite im Bereich »von« »bis« durch. Sprungbefehle werden dabei berücksichtigt.

AUTO Start,Schritt
...ergibt eine automatische Zeilenvorgabe mit der ersten Zeilennummer »Start« und der Schrittweite Schritt.

```
AUTO100,10
```

ergibt die Zeilenvorgabe 100 und erhöht nach jeder Programmzeile um 10. Die Bestätigung einer Zeilennummer (<RETURN>) ohne Eingaben, beendet diesen Modus.

DATA Start,Schritt
...wie »AUTO«, nur wird zusätzlich »DATA« nach der Zeilennummer ausgegeben.

DEL von-bis
...löscht einen Bereich »von« »bis« aus dem Programmspeicher.

```
DEL120-150
```

löscht alle Programmzeilen von 120 bis inkl. 150.

REPLACE Zeichenkette
...dient zum Ersetzen beliebiger Zeichenketten. Nehmen wir als Programmbeispiel:

```
100 A=100
110 A$="HALLO"
```

Um Strings zu ersetzen, geben Sie ein:

```
REPLACE"HALLO"
```

wenn Sie danach mit <RETURN> bestätigen, erscheint am Bildschirm

```
bY:
"
```

und der Cursor steht hinter den Anführungszeichen. Geben Sie hier die Zeichenkette ein und bestätigen Sie mit <RETURN>. Die Routine kann auch Befehle tauschen:

```
REPLACE A=100
BY:
B=999
```

Nach <RETURN> steht dieser Befehl in Zeile 100.

FIND Zeichenkette
...sucht nach »Zeichenkette« und listet die entsprechenden Zeilen. Wie oben wird der »Anführungszeichen-Modus« unterstützt.

SEARCH Zeichenkette
...wie »FIND«, nur wird eine Programmseite ab der ersten gefundenen Zeichenkette gelistet.

DUMP
...listet alle Variablen mit ihren Werten.

HELP
...listet die zuletzt vom Programm bearbeitete Zeile und zeigt die Stelle, an der das Betriebssystem bei einem Fehler abgebrochen hat revers.

TRACE Zeilennummer
...startet das Programm und zeigt die aktive Zeile in der rechten oberen Bildschirmecke an. Zusätzlich wird nach jedem auf eine der Tasten <SHIFT>, <CTRL> oder <CBM> gewartet. Falls das Programm fortlaufend betrachtet werden soll ein Tip: verwenden Sie <SHIFT LOCK>.

OLD
...rettet ein mit >NEW« gelöschtes Programm.

Floppy

Die Routinen DLOAD, DVERIFY, DAPPEND und DMERGE sind sechsmal schneller als die Standardroutinen, arbeiten aber nur mit der Floppy 1541 zusammen. Bei einer anderen Floppy s.a. »allgemeine Routinen« »MERGE« und »APPEND«.

DIR oder Pflhoch
...zeigt das Inhaltsverzeichnis der Diskettenstation

. Diskbefehl
...ohne Parameter eingegeben liest den Fehlerkanal aus. Mit Parametern wird ein Befehl zur Floppy übertragen:

```
.s:test
```

löscht das File »TEST« auf Diskette.

DLOAD Dateiname
...lädt die Datei »Dateiname« von Diskette

BLOAD Dateiname
...lädt ein Maschinenprogramm an die dafür vorgesehene Speicherposition.

DMERGE Dateiname
...fügt ein Basic-Programm von Diskette in ein schon bestehendes im Speicher.

DAPPEND Dateiname
...lädt ein Basic-Programm von Diskette hinter einem.

DSAVE Dateiname
...transferiert das Programm im Speicher unter dem Namen »Dateiname« auf Diskette.

DBSAVE von-bis,Dateiname
...speichert nur die Zeilen »von« »bis« als Programm auf Disk.

DVERIFY Dateiname
...vergleicht das Programm »Dateiname« von Diskette mit dem im Speicher.

Tape

Die Kassettenroutinen sind sechsmal schneller als die Standardroutinen.

TLOAD Dateiname
...lädt die Datei »Dateiname« von Band

TMERGE Dateiname
...fügt ein Basic-Programm von Band in ein schon bestehendes im Speicher.

TAPPEND Dateiname
...lädt ein Basic-Programm von Band hinter eines im Speicher.

TSAVE Dateiname
...transferiert das Programm im Speicher unter dem Namen »Dateiname« auf Band.

TBSAVE von-bis,Dateiname
...speichert nur die Zeilen »von« »bis« als Programm auf Band.

TVERIFY Dateiname
...vergleicht das Programm »Dateiname« auf Band mit dem im Speicher.

Zum Schluß noch ein Tip: Schalten Sie alle Module oder Floppy-Speeder aus, da sonst einige Funktionen nicht oder nur fehlerhaft ausgeführt werden. (gr)

Kurzinfo: ExtEditor

Programmart: Programmierhilfe
Laden: LOAD"EXTEDITOR",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 31
Programmautor: R. T. Splinter

Kleine Tools für Anfänger und Profis

Die Tool-Kiste

Mit unseren kleinen Tricks wird das Handling des C64 einfacher und komfortabler. Vom Speichern einzelner Zeilenbereiche bis zum Reset-Schutz spannt sich der Bogen.

Als erstes (mini-) Programm stellen wir Ihnen eine Routine vor, die es erlaubt, einzelne Zeilenabschnitte (z.B. Unterprogramme) aus einem Basic-Programm zu speichern. Davon hat sie auch den Namen »Clip« erhalten. Die Routine selbst liegt dabei im Bereich von $033C bis $03CB, also im Kassettenpuffer. Daher muß sie unmittelbar geladen werden:

```
LOAD"CLIP",8,1
```

Da der Kassettenpuffer außer bei Tape-Operationen nicht vom Betriebssystem verwendet wird, kann das Basic-Programm vorher im Speicher stehen, oder danach geladen werden. Lediglich ein RESET zerstört diesen Bereich. Verwenden Sie ebenfalls kein »NEW«, da dieser Befehl das Basic-Programm unsichtbar macht. Wenn Sie Clip einmal im Speicher haben, stehen vier Varianten der Speicherung zur Verfügung:

SYS828,"name",Zeile

... speichert nur »Zeile« unter dem Direktory-Eintrag »name« und muß numerisch eingegeben werden.

```
SYS828,"TEST",100
```

speichert nur die Zeile 100 unter dem Namen »TEST«.

SYS828,"name",Zeile-

... speichert das Basic-Programm ab »Zeile« unter dem Direktoryeintrag »name«.

```
SYS828,"TEST1",100-
```

speichert das Basic-Programm ab Zeile 100 unter dem Namen »TEST1«.

SYS828,"name",Anfang-Ende

... speichert aus einem Basic-Programm die Zeilen »Anfang« bis »Ende« unter dem Direktory-Eintrag »name«.

```
SYS828,"TEST2",100-200
```

speichert aus einem Basic-Programm die Zeilen 100 bis 200 unter dem Namen »TEST2«.

SYS828,"name",-Ende

... speichert ein Basic-Programm »Anfang« bis zur Zeile »Ende« unter dem Direktory-Eintrag »name«.

```
SYS828,"TEST3",-200
```

speichert ein Basic-Programm bis zur Zeile 200 unter dem Namen »TEST3«.

Die Parameterverarbeitung entspricht der des LIST-Befehls. Die Routine verwendet die normalen Pointer, wie sie auch vom Betriebssystem angewandt werden. Daher läßt sich auch die Geräteadresse ändern:

Mit »POKE186,Geräteadresse« und anschließendem »SYS828,Parameter« leiten Sie die Ausgabe zum gewünschten Gerät. Verwenden Sie aber auf keinen Fall die Datasette, da Clip sonst überschrieben wird.

Kurzinfo: Clip

Programmart: Basic-Tool zur Speicherung von Programmabschnitten
Laden: LOAD"CLIP",8,1
Benötigte Blocks: 1
Programmautor: Hermann Schinagl

INPUT ohne Fragezeichen

Eine von vielen Möglichkeiten, das lästige Fragezeichen beim INPUT-Befehl wegzubekommen, ist die folgende:

```
OPEN 1,0 : INPUT #1,A$ : CLOSE 1
```

Der kleinste Sprite-Editor

Mit diesem Zweizeiler läßt sich ein Sprite entwerfen. Die Sprite-Daten werden dabei automatisch auf dem Bildschirm ausgegeben.

Wenn das Programm geladen beziehungsweise abgetippt worden ist, darf es zunächst nicht mit RUN <RETURN> gestartet werden. Zuerst ist im Direktmodus mit der SHIFT CLEAR/HOME-Taste der Bildschirm zu löschen. Jetzt kann, angefangen in der linken oberen Bildschirmecke, das Sprite entworfen werden:

<*> setzt einen Punkt
<SPACE> löscht beziehungsweise überspringt einen Punkt

Das Eingabefeld hat eine Größe von 21 Spalten und 24 Zeilen.

Sind Sie mit Ihrem Kunstwerk zufrieden und wollen die Sprite-Daten ausgeben, ist der Cursor am Anfang der vorletzten Bildschirmzeile zu positionieren.
RUN <RETURN> an dieser Stelle startet schließlich den Sprite-Editor, der die Daten errechnet und auf dem Bildschirm ausgibt.

```
1 FOR I=0 TO 2: A=0: FOR N=0 TO 7: A=A-2↑(7-N)*
(PEEK (1024+N+I*8) = 42): NEXT
2 PRINT A;: NEXT: PRINT: X=X+1: IF X<21 THEN 1
```

(Daniel Wicker)

Token-Finder

Jede Basic-Erweiterung hat auch ihren eigenen Wortschatz. Klar, daß es schier unmöglich ist, alle Befehle auswendig zu wissen. Also kramt man die Programmbeschreibung heraus und schreibt die einzelnen Kommandos heraus. Eine weitaus komfortablere Methode alle Anweisungen auf einen Blick zu erhalten, bietet der Token-Finder. Er sucht alle verwendeten Befehle und schreibt sie in eine Programmliste (Abb. 1), zusammen mit der verwendeten Codenummer. Die Zeilennummern entsprechen dabei den Dezimalwerten, daneben wird der Hexadezimalcode und das Token (Befehlscode des Interpreters) eingetragen. Später, beim LISTen wird das Token automatisch vom Betriebssystem in Klartext gewandelt.

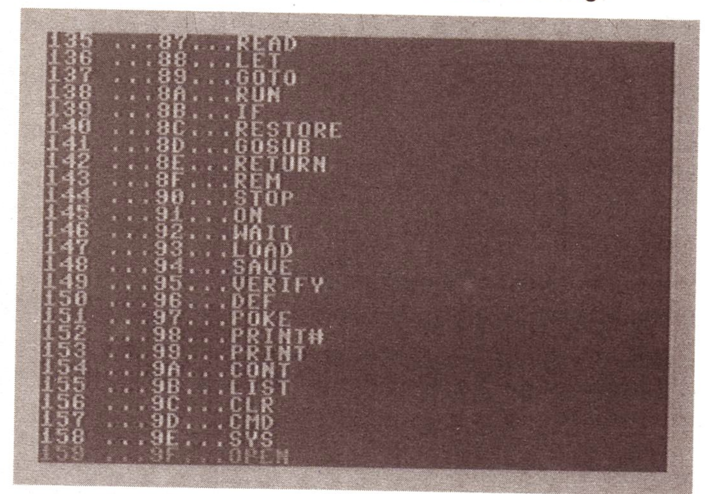
Geladen wird das Basic-Tool mit

```
LOAD"TOKEN-FINDER",8
```

und gestartet mit RUN. Sie sehen jetzt am Bildschirm oben jeweils die (Zeilen-) Nummer des Tokens, das gerade bearbeitet wird. Wenn die Ausführung beendet ist, löscht sich das eigentliche Programm selbst. Da zum Schluß versucht wird, eine gelöschte Basic-Zeile anzuspringen, endet die Routine mit einem »UNDEF'D STATEMENT ERROR«. Dieser etwas ungewöhnliche Schluß ist aber kein echter Fehler.

Wenn Sie den Token-Finder nach dem Laden mit LIST betrachten, verstehen Sie die Funktionsweise:

Beim ersten Programmstart wird (Zeile 400) in die Speicherstelle »2« der Wert 127 gePOKEd. Dazu muß man wissen, daß alle Token größer als 127 sind. In Zeile 410 wird dieser Wert wieder ausgelesen, um eins erhöht und in die Variable »Z« übernommen. Wenn in Zeile 420 dieser Wert gleich 256 ist, ist die maximale Token-Anzahl überschritten, es wird zum zweiten Programmteil (ab Zeile 550) verzweigt.



[1] Der Token-Finder zeigt alle Basic-Befehle und Anweisungen

Ansonsten zerlegen Zeile 440 bis 480 diesen Wert in eine Hexadezimalzahl. Dieser Wert wird zusammen mit der Variablen »Z« und einem Platzhalter auf den Bildschirm gebracht (Zeile 490). Danach schreibt das Programm »RUN 410« auf den Screen. »Z« wird wieder in Speicherstelle »2« gePOKEd. Zeile 520 bis 530 schreibt in den Tastaturpuffer »Cursor an erste Bildschirmposition« und <RETURN>. Danach nochmals <RETURN>. Unmittelbar danach wird die aktuelle Länge auf »3« gesetzt. Wenn jetzt in Zeile 540 das Programm beendet ist, stehen im Tastaturpuffer noch Werte und das Betriebssystem führt diese auch aus. Damit wird die erste Zeile mit <RETURN> in den Speicher übernommen, und anschließend »RUN410« ausgeführt. Damit startet der Token-Finder wieder, bis die Zeilennummer 256 lautet. Bei dieser Zahl verzweigt er zu Zeile 550. Hier wird zuerst der Programmstart ermittelt und danach die 256 Token in den Platzhalter gePOKEd (Zeile 560 bis 580). Die Zeilen 590 bis 600 sorgen für das Löschen des eigentlichen Programms. Da auch diese Routine sich selbst aus dem Programm löscht, kommt es zwangsläufig zum »UNDEF'D STATEMENT ERROR«.

Kurzinfo: Token-Finder

Programmart: zeigt Basic-Befehlssatz
Laden: LOAD"TOKEN-FINDER",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 2

Das Zeilenlineal

Kennen Sie diese Situation? Man hat ein Listing aus dem 64'er-Magazin abgetippt und irgendwo einen Tippfehler gemacht. Sie lassen sich also mit dem LIST-Befehl die ersten Programmzeilen auf dem Bildschirm ausgeben. Nun wird Zeile für Zeile mit dem Listing im Heft verglichen. Wie leicht kann man da um eine Zeile verrutschen. Auf der Heftseite läßt sich ein Lineal anlegen, aber am Bildschirm?

Hier hilft Ihnen der Zeilenmarker (Abb. 2) weiter. Er hebt die gerade bearbeitete Zeile farblich hervor. Eine nicht zu unterschätzende Hilfe. Geladen wird mit

```
LOAD"ZEILENMARKER",8
```

und gestartet mit Run. Danach schiebt sich das Programm in den Bereich ab 49152 und wird aktiviert. Wenn Sie es nicht mehr brauchen, drücken Sie einfach <RUN/STOP RESTORE>. Danach kann der Zeilenmarker mit SYS49156 wieder eingeschaltet werden. Falls Ihnen die Farben nicht mehr gefallen, ändern Sie mit

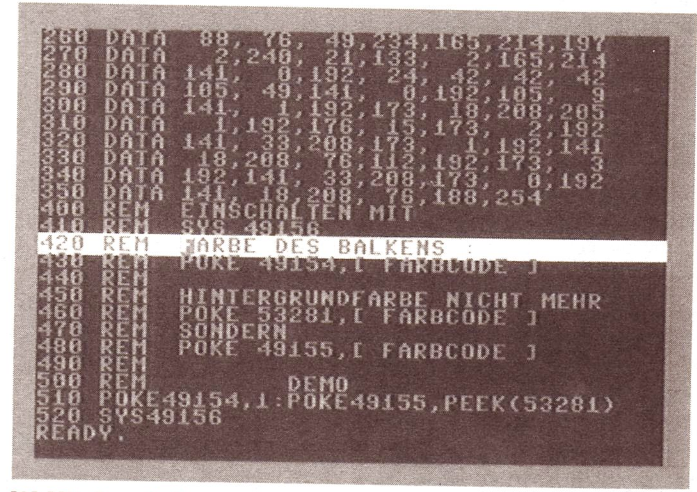
```
POKE49154,x
```

die Farbe des Balkens. »x« kann dabei die übliche Wertereihe von Null bis Fünfzehn haben. Die Farben entsprechen denen des C64.

```
POKE49155,x
```

wechselt die Hintergrundfarbe.

Achtung: Dieses Tool arbeitet im Interrupt. Einige Steckmodule können daher Fehlfunktionen hervorrufen.



[2] Mit dem Zeilenmarker sticht die Cursor-Zeile förmlich ins Auge

Kurzinfo: Zeilenmarker

Programmart: markiert die aktuelle Cursor-Zeile
Laden: LOAD"ZEILENMARKER",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 4
Programmautor: Gregor Alexander Piel

RESET-Schutz

Normalerweise kann man durch einen zusätzlich eingebauten RESET-Taster jedes beliebige Programm abbrechen und den Computer in den Einschaltzustand versetzen. Eine Kennung in den richtigen Speicherbereich geschrieben verhindert dies. Sie ist eigentlich dafür gedacht, ein Programmodul im Expansion-Port gleich nach dem Einschalten zu starten, funktioniert aber auch ohne Modul. Dazu muß ab Speicherstelle $8004 die Zeichenfolge »CBM80« stehen. Die vier Speicherstellen davor müssen zwei Pointer enthalten, die auf die eigene RESTORE- und RESET-Routine zeigen. Wenn Sie nun die Speicherstellen auf die entsprechenden Einsprünge des Betriebssystems deuten lassen, z.B. auf den Start des RUN-Befehls, startet ein RESET oder <RUN/STOP RESTORE> das Basic-Programm neu. Genau diese Aufgabe übernimmt der RESET-Schutz, wenn er zu Beginn eines Basic-Programms steht. Geladen wird er mit

```
LOAD"RESET-SCHUTZ",8
```

Wenn er gestartet wird, überträgt er ein kleines, 65 Blocks langes Maschinenprogramm in den Speicher ab Position 53000 und initialisiert es.

Wenn Sie den RESET-Schutz an den Anfang eines eigenen Programms einbauen, kann kein RESET mehr ausgeführt werden. Der C 64 führt dann nach jedem Auslösen automatisch ein RUN aus.

Kurzinfo: RESET-Schutz

Programmart: verhindert Software-RESET
Laden: LOAD"RESET-SCHUTZ",8
Starten: nach dem Laden in eigenes Programm einbauen RUN eingeben
Benötigte Blocks: 2
Programmautor: Jochen Hammerschmidt

Lightpen für Amica Paint

Wenn Sie daheim noch einen Lightpen besitzen, ist es auch sinnvoll, ihn für das Malprogramm Amica Paint zu verwenden. Mit dem Eingabetreiber »INPUT.PEN« ist das jetzt möglich. Dazu gehen Sie in folgender Reihenfolge vor:

Verwenden Sie eine Sicherheitskopie von Amica Paint, bei der Sie »EINGABEGERAETE« nicht mitkopieren. Kopieren Sie anstelle dessen »EINGABEGERAETE« von der beiliegenden Diskette. Zusätzlich kopieren Sie »INPUT.PEN« mit auf Ihre Arbeitsdiskette.

Achtung: Verwenden Sie dazu ein Kopierprogramm, das in der Lage ist, Maschinenprogramme zu kopieren.

Wenn Sie jetzt Amica Paint starten, finden Sie einen neuen Menüpunkt zur Installation des Lightpens. Er ist in Joystick-Port 1 einzustöpseln. Nach dem eigentlichen Start sollten Sie unter »Seite löschen« einen hellen Hintergrund wählen (weiß, <S> bis <L>), da der Pen sonst nicht anspricht. Natürlich muß die Zeichenfarbe einen anderen Wert enthalten (<F> bis <Z>). Mit <+> und <-> läßt sich die Penspitze punktgenau justieren.

Achtung: Die Funktionen »Muster editieren«, »Stricheln editieren« und »Farbprioritäten setzen«. Die anderen Funktionen lassen sich uneingeschränkt nutzen. (gr)

Kurzinfo: Lightpen für Amica Paint

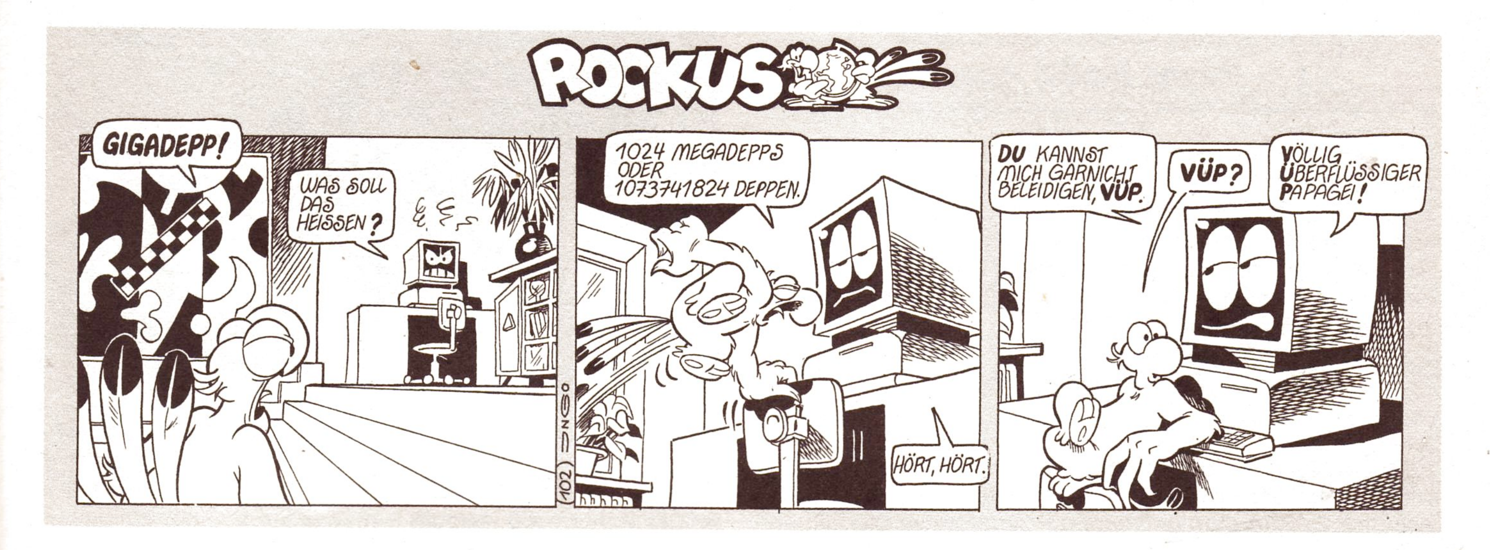
Programmart: Erweiterung der Eingabegeräte
Benötigte Blocks: 2 und 5 für Eingabegeräte
Programmautor: Steffen Thiel

Der Mini-Autostart

Wer zu faul ist, jedesmal »RUN« einzutippen, wenn ein Programm von der Floppy geladen wurde, kann jetzt aufatmen: Ein kleiner Trick macht dies automatisch.

Wie Sie vielleicht wissen, bewirkt ein SHIFT-RUN/STOP-Tastendruck ein Laden des nächsten Programms von Datasette mit automatischem RUN danach. Diesen Umstand können sich die Floppy-Besitzer zunutze machen.

Wenn Sie ein Programm laden möchten, tippen Sie ganz normal »LOAD"Programmname",8« ein und setzen dahinter noch einen Doppelpunkt. Drücken Sie nun <SHIFT-RUN/STOP>. Auf dem Bildschirm erscheint hinter dem Doppelpunkt noch ein LOAD-Befehl (dieser wird vom C 64 jedoch ignoriert). Wenn das Programm fertig geladen ist, führt der Computer ein RUN aus. (Bernd Roggendorf)



Mg version 1.0 – Modulgenerator für EPROMs

Tools à la Carte

Bauen Sie Ihre eigene EPROM-Tool-Sammlung auf und starten Sie die einzelnen per Tastendruck aus einem Menü – mit dem Modulgenerator ist das ein Kinderspiel. Er nimmt Ihnen komplizierte Programmierarbeit ab und erlaubt sogar die Nachbearbeitung fertiger Module.

Ein gewöhnlicher Generator wandelt jedes einzelne Programm in eine brennfertige Form. Dabei müssen alle Adressen angegeben werden, die sich auf die Programme und die EPROMs beziehen – ein ziemlich komplexer Vorgang. Nicht so bei unserem Modulgenerator, denn er erkennt die Startadresse und Endadresse automatisch und schweißt alle Programme zu einem Paket zusammen. Schließlich wird noch ein Auswahlmenü hinzugefügt.

Überschreitet ein Programm die Speichergrenze des ersten EPROMs, wird der Rest davon für das nächste vorbereitet.

Aber auch eine andere Startadresse läßt sich verwenden. Dazu existiert ein eigener Menüpunkt. Sollen einige Änderungen an einem fertigen Modul vorgenommen werden, z.B. Startadressen neu eingeben, muß man nur das alte Modulprogramm von Diskette laden, schon kann's mit dem Editieren losgehen. Dies gilt allerdings nur für Module, die mit dem »Modul-Generator C 64« entwickelt wurden. Geladen wird mit

```
LOAD"MG VERSION 1.0",8
```

und gestartet mit RUN. Folgende Menüpunkte stehen danach zur Verfügung (Abb. 1). Die Anwahl geschieht per Tastendruck:

<1> Neues Modul generieren

Nach Anwahl dieses Menüpunkts wird der Speicherplatz ohne Ihr Zutun für die Generierung neuer Module vorbereitet. Eventuell zuvor bearbeitete und noch im Speicher befindliche Module werden gelöscht. Angezeigt wird der nunmehr wieder voll verfügbare Modulspeicherplatz in den untersten Zeilen des Hauptmenüs. Diese Option muß zu Beginn jeder Sitzung angewählt werden.

<2> Programm laden

Eine Funktion, die die vorher zu definierenden Programme von Diskette lädt. Danach wird die Eingabe eines Menütextes erwartet. Er wird dem geladenen Programm im Modulmenü zugewiesen. Es stehen maximal 30 Schreibstellen zur Verfügung. Der Textzeile wird automatisch eine Funktionstaste für den Programmaufruf im fertigen Modul zugeordnet. Die Reihenfolge ist dabei fest vorgegeben:

<F1>, <F2>, <F3>, <F4>, <F5>, <F6>, <F7>, <F8>.

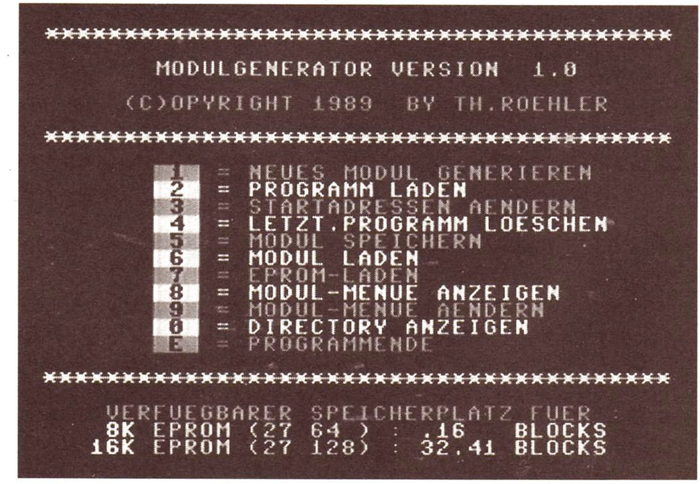
Sollte die Startadresse des geladenen Programms nicht mit dem Basic-Anfang (2049) identisch sein, kann die Startadresse oder ein Startmodus bestimmt werden (Abb. 2). Sie haben zur Auswahl:

RUN – startet später das entsprechende Programm mit RUN aus dem Modul.

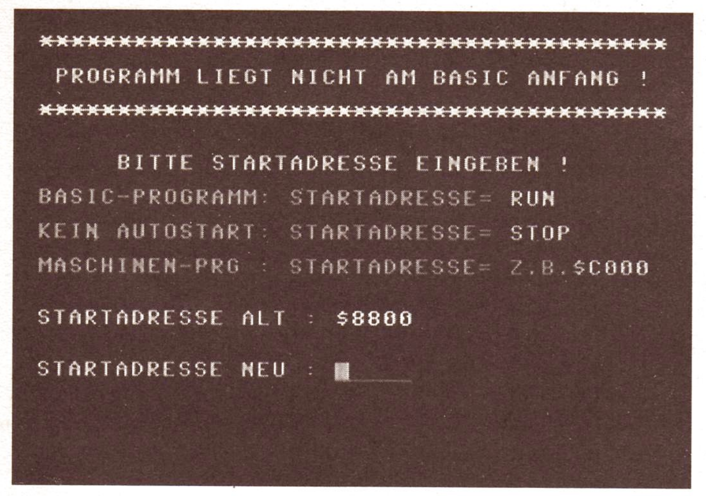
STOP – bei Anwahl dieses Programms erfolgt im fertigen Modul kein Autostart, d.h. der C 64 befindet sich nach Übernahme des Programms aus dem Modul im normalen Einschaltmodus.

$XXXX – startet ein Maschinenprogramm nach der Übernahme aus dem Modul automatisch mit der einzugebenden Einsprungsadresse im Dezimal- oder Hexadezimalcode (entspricht dem SYS-Befehl).

Nach Abschluß dieses Programmpunkts wird im Hauptmenü die Anzeige des noch verfügbaren Speicherplatzes aktualisiert. Wird die Speicherkapazität eines 8K-EPROMs überschritten, erfolgt die Berechnung und Anzeige des freien Speicherplatzes nur noch auf der Basis eines 16K-EPROMs. Sollte auch dieser überschritten werden, erfolgt eine Meldung und das Programm wird nicht in das Modul eingebunden. Mit <RUN/STOP> kann abgebrochen werden.



[1] Das Hauptmenü stellt elf Optionen zur Verfügung



[2] Der Modulgenerator erkennt Maschinenprogramme und erfragt die Startanweisung

<3> Startadresse ändern

Dieser Programmpunkt ermöglicht die nachträgliche Änderung von Startadressen und Startmodi bei Programmen, die bereits zu einem Modul zusammengefaßt sind.

Zunächst wird das Modulmenü ausgegeben. Das gewünschte Programm wird mit der zugeordneten Funktionstaste gewählt und anschließend mit den neuen Daten versehen (siehe Menüpunkt 2).

<4> Letztes Programm löschen

Das im Modulmenü zuletzt aufgeführte Programm wird mit dieser Funktion gelöscht. Vor dem Löschen erfolgt eine Sicherheitsabfrage:

»SIND SIE SICHER? (J/N)«

Durch die Angaben von <N> wird der Programmpunkt ohne Konsequenzen verlassen. Bei <J> wird der vom Programm belegte Speicherplatz wieder freigegeben und das aktuelle Menü angezeigt. Mit einer beliebigen Taste erreichen Sie wieder das Hauptmenü.

<5> Modul speichern

Dieser Menüpunkt verlangt zuerst den Namen des Moduls und speichert anschließend auf Diskette. Ist dies fehlerfrei geschehen, wird das Modul gelöscht. Mit <RUN/STOP> wird dieser Punkt abgebrochen.

<6> Modul laden

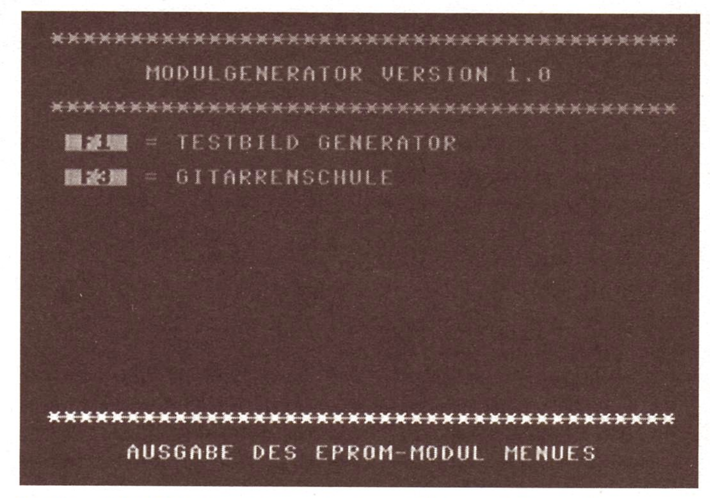
Nach Aufruf dieser Option wird der Name des zu ladenden Moduls verlangt. Sollte es nicht mit dem »Modul-Generator C64« zusammengestellt sein, erfolgt eine entsprechende Fehlermeldung und es wird nicht in den Arbeitsspeicher übernommen. Ansonsten wird geladen und zur weiteren Bearbeitung ins Hauptmenü zurückgekehrt.

<7> EPROM laden

Dieser Programmpunkt stellt eine Funktion zur Verfügung, die sonst nur von EPROM-Programmiergeräten geboten wird, nämlich das Auslesen einer EPROM-Karte. Sie muß natürlich auszulesende EPROM enthalten. Die Karte muß sich ein- bzw. ausschalten lassen. Danach fragt das Programm nach dem EPROM-Typ. Nun muß die Karte eingeschaltet und nach dem Lesen des EPROMs wieder ausgeschaltet werden. Es lassen sich nur Module bearbeiten, die mit dem Modulgenerator zusammengestellt wurden.

<8> Modulmenü anzeigen

Nach dem Anwählen erscheint das Menü des Moduls (muß im Speicher sein) auf dem Bildschirm (Abb. 3).



[3] Das EPROM-Menü kann jederzeit überprüft werden

<9> Modulmenü ändern

Mit dieser Funktion lassen sich nachträgliche Änderungen im Modulsmenü. Dazu erscheint zuerst das bestehende Menü auf dem Bildschirm. Dann wird auf die Eingabe der zu ändernden Textzeile gewartet. Soll die Kopfzeile des Menüs geändert werden, betätigt man anstatt einer Funktionstaste <K>. Das Programm gibt den alten Text aus und fragt den neuen ab. Nach der Eingabe und Bestätigung mit <RETURN> wird der alte Text durch den neuen ersetzt und ins Modulpaket übernommen.

<0> Direktory anzeigen

Dieser Menüpunkt zeigt das Inhaltsverzeichnis der eingelegten Diskette am Bildschirm.

<E> Programmende

Achtung: Ein im Speicher vorhandener Datensatz wird nach Neustart des Generators gelöscht.

Jede Funktion (außer Eingaben) des Modulgenerators kann mit der Taste <RUN/STOP> zum Hauptmenü hin abgebrochen werden. Vor dem Verlassen des Programms empfiehlt es sich, ein im Speicher vorhandenes Modulpaket abzuspeichern, da nach einem Neustart des Modulgenerators der Arbeitsspeicher wieder freigegeben wird. (gr)

Kurzinfo: Modulgenerator Version 1.0

Programmart: EPROM-Softwaregenerator
Laden: LOAD "MG VERSION 1.0",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 34
Programmautor: Thomas Röhler

Schneller als der Schall

Wer mit dem C64 arbeitet, ärgert sich ständig über die mäßige Geschwindigkeit der Floppy 1541. Damit ist Schluß: Wir zeigen, wie Sie sich einen leistungsfähigen Speeder für wenig Geld bauen.

Hypra-Speed ist ein paralleler Beschleuniger für den User-Port, mit einem Potential, das man bisher nur von teuren, kommerziellen Beschleunigern her kennt. Beispielsweise enthält er zusätzliche Befehle. Eine sinnvolle Belegung der Funktionstasten runden den Bedienungskomfort ab. Über die eingebaute Centronics-Schnittstelle lassen sich entsprechende Drucker (Centronics-Schnittstelle) betreiben. Zusätzlich können Sie 40 Spuren formatieren. Dadurch erhöht sich die Speicherkapazität einer Diskette um 85 Blöcke.

Hypra-Speed ist ein paralleler Beschleuniger. Beim normalen seriellen Verfahren sendet der C64 ein Byte nicht komplett zur Floppy. Er schickt es Bitweise. Damit sind acht Sendedaten zur Übertragung nötig. Im Unterschied dazu wird bei paralleler Datenübertagung immer ein gesamtes Byte übermittelt. Dadurch sind zwar mehr Leitungen nötig, aber bei gleicher Übertragungsrate wird nur ein achtel der Zeit benötigt. Da zusätzlich die Fehlererkennung einfacher wird, ist die tatsächliche Geschwindigkeit noch höher (ca. Faktor 10). Sie benötigen also ein spezielles Kabel für die Verbindung der Floppy mit dem User-Port. Dazu später mehr. Zusätzlich muß das Betriebssystem im C64 und im Laufwerk gegen ein neues ausgetauscht werden. Dafür bereiten Sie eine Arbeitsdiskette vor. Legen Sie eine fabrikfrische Diskette ins Laufwerk und formatieren sie z.B. mit:

```
OPEN1,8,15:PRINT#1,"N:HYPRA,HF":CLOSE1
```

Wenn Sie diese Eingabe mit <RETURN> bestätigen, verschwindet der Cursor solange, bis die Diskette formatiert ist. Danach legen Sie die beiliegende Diskette ein, und laden

```
LOAD "HYP.PACKED",8
```

Im Anschluß an den Ladevorgang legen Sie wieder die vorher formatierte Disk ein und speichern

```
SAVE "HYP.PACKED",8
```

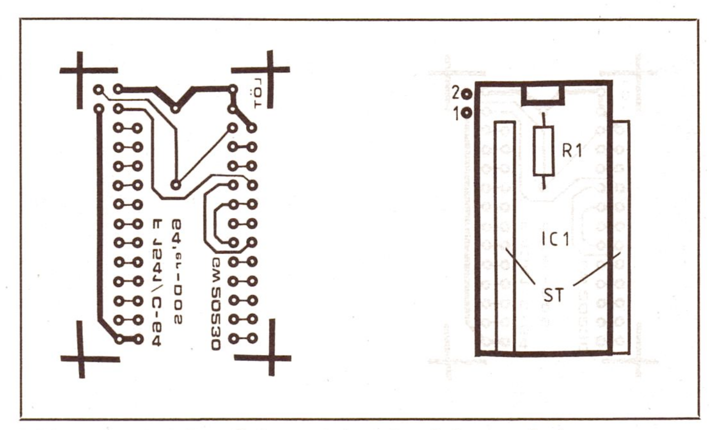
Danach starten mit RUN. Das Programm erzeugt drei Files auf Diskette, »DELTA.C64«, »DELTA.DOS« und »CATCH!«. Die drei Programme werden von dem Install-Programm benötigt, das Sie ebenfalls von der beiliegenden Diskette auf die Arbeitsdiskette kopieren:

```
LOAD"INSTALL",8
```

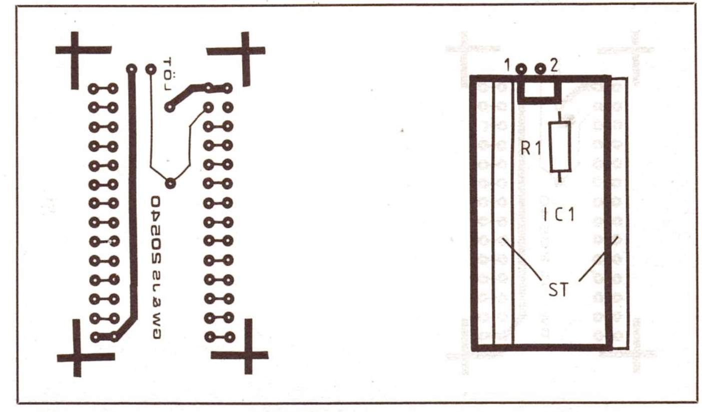
lädt das Installationsprogramm von der beiliegenden Diskette und

```
SAVE"INSTALL",8
```

speichert es auf die Arbeitsdiskette, die Sie natürlich vorher einlegen sollten. Startet man dieses Programm nun mit RUN, werden zunächst die zuvor erzeugten Files nachgeladen und ein Menü erscheint auf dem Bildschirm. Hier müssen Sie eingeben, in welchen Computertyp Hypra-Speed eingebaut



[1] Layout und Bestückungsplan der Adapterplatine (16-KByte-Betriebssystemumschaltung). Achtung: Das Layout ist spiegelverkehrt im Verhältnis 1:1.

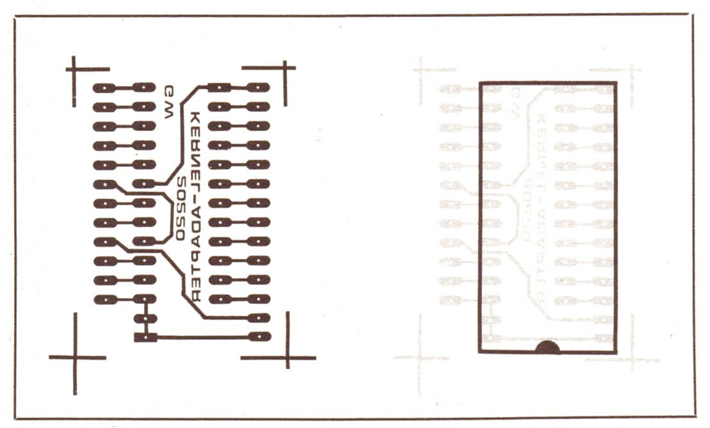


[2] Layout und Bestückungsplan der Adapterplatine (32-KByte-Betriebssystemumschaltung). Achtung: Das Layout ist spiegelverkehrt im Verhältnis 1:1.

werden soll (Tabelle 1). Ist das geschehen, wird das Betriebssystem des C64 bzw. C128 ausgelesen, geändert und unter »Kernal-ROM« auf Diskette gespeichert. Dieser Vorgang dauert etwa 47 Sekunden. Anschließend erscheint ein weiteres Menü, in dem der EPROM-Typ für die Floppy anzugeben ist. Welcher Typ benötigt wird, zeigt Tabelle 1. Nun wird das Betriebssystem der Floppy ausgelesen, ebenfalls geändert und unter »Floppy-ROM« auf Diskette gespeichert. Das Auslesen dauert etwa vier Minuten. Beide Files sind mit einem EPROMer Ihrer Wahl in die entsprechenden EPROMs zu brennen (Tabelle 1). Sollten Sie keinen EPROMer besitzen, können Sie die EPROMs sowie alle erforderlichen Teile bei der unter diesem Artikel angegebenen Bezugsadresse bestellen. Da die im C64 bzw. in der Floppy eingebauten ROM-Bausteine teilweise nicht PIN-kompatibel zu den EPROMs sind, benötigt man unter Umständen noch zwei sog. Adapterplatinen. Das Layout dazu finden Sie im Verhältnis 1:1 in Abb. 1 (8-KByte-EPROM; Betriebssystemumschaltversion), Abb. 2 (32-KByte-EPROM; Betriebssystemumschaltversion) oder Abb. 3 (8-KByte-EPROM; Normalversion). Für welche der drei angebotenen Platinen Sie sich entscheiden, bleibt Ihnen überlassen. Ob Sie eine Adapterplatine brauchen oder nicht, entnehmen Sie ebenfalls Tabelle 1. Bestückt wird die Platine mit (einem Widerstand R1 = 10K für die Betriebssystemumschaltversionen) einer 28poligen IC-Fassung und zwei 12poligen Stiftleisten (von unten). Von den Stiftleisten gibt es zwei verschiedene Sorten, dicke und dünne. Nehmen Sie die dünnen Stiftleisten. Die dicken sind zwar preiswerter, zerstören jedoch die IC-Fassung im C64.

Tabelle 1 enthält eine zusätzliche Spalte, sie gibt den zu verwendenden EPROM-Typ für eine Betriebssystemumschaltplatine an. Wenn Sie diese verwenden, ist noch der Schalter (ein/aus) einzulöten. Sie können dann entweder mit dem originalen Betriebssystem oder mit Hypra-Speed arbeiten, und zwar sowohl im Computer als auch in der Floppy. Da ein EPROM zwei Betriebssysteme enthält, sind das originale Betriebssystem und Hypra-Speed nacheinander zu brennen. Steht der Schalter auf »aus«, wird das untere Betriebssystem beziehungsweise die untere EPROM-Bank selektiert.

Ein Wort noch zu den IC-Fassungen: Um den Adaptersockel oder ein EPROM einzubauen, muß der entsprechende ROM-Baustein natürlich gesockelt sein. Leider ist das nicht immer der Fall. Sollten in Ihrem Computer oder Ihrer Floppy die zu wechselnden Bausteine nicht gesockelt sein, müssen sie vorsichtig ausgelötet und durch eine IC-Fassung ersetzt werden. Da dies erfahrungsgemäß eine etwas heikle Angelegenheit ist, empfehlen wir dem nicht so versierten Bastler, diese Arbeit von einem Fachmann durchführen zu lassen (z.B. im Radio- und Fernsehfachhandel). Bedenken Sie: Der Ärger ist groß, wenn beim Anlöten Leiterbahnen abreißen und nachher nichts mehr funktioniert.



[3] Layout und Bestückungsplan der Adapterplatine (8-KByte-Betriebssystemumschaltung). Achtung: Das Layout ist spiegelverkehrt im Verhältnis 1:1.

Tabelle 1: Welches EPROM?

	Steckplatz für EPROM		Benötigter EPROMtyp bei Verwendung der Betriebssystemumschaltplatine	Zwischensockel
C64 (alt; Brotkasten)	U4 2764	mit Adapterplatine	27128	
C64II (weiß; flach)	U4 2764	mit Adapterplatine	27128	
C64 (kleine Platine)	U4 27128	ohne Adapterplatine	27256	
C128	U32 27128	ohne Adapterplatine	27256	
1541	UB4 2764	mit Adapterplatine	27128	UC3
1541 C	UA2 27128	ohne Adapterplatine	27256	UC1
1541 II	U4 27128	ohne Adapterplatine	27256	U6

Tabelle:2 Verdrahtungsschema des Parallelkabels

Die Pinnummern des Zwischensockels beziehen sich nicht auf die IC-Pins, sondern auf die zehn für das Kabel vorgesehenen Lötaugen.

Zwischensockel		User-Port-Stecker
1	-------	M
2	-------	L
3	-------	K
4	-------	J
5	-------	C
6	-------	D
7	-------	E
8	-------	F
9	-------	5
10	-------	H

Mit dem Brennen der EPROMs und dem eventuellen Anfertigen der Adaptersockel ist die Hardwarebastelei noch nicht getan. Was fehlt, ist das parallele Kabel. Den Verdrahtungsplan dazu zeigt Tabelle 2 und das Layout für den Zwischensockel Abb. 4.

Achtung: Beim Anlöten des Kabels an den 40poligen IC-Zwischensockel bzw. an den User-Port-Stecker ist größte Vorsicht geboten. Ein Kurzschluß kann die Elektronik des Diskettenlaufwerks oder des C64 zerstören.

Ist das Kabel fertig, muß der 40polige IC-Zwischensockel unter den 6522-Baustein in die Floppy eingebaut werden.

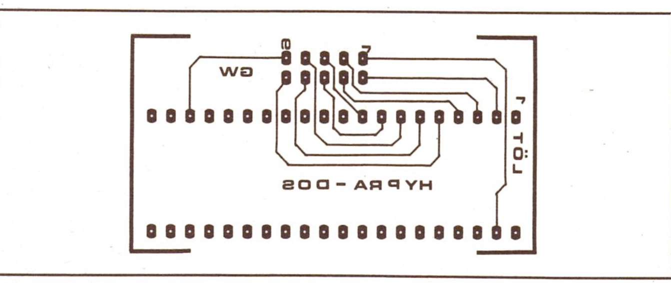
Wichtig: Soll Hypra-Speed in die 1541 II oder 1541c eingebaut werden, ist Pin 2 vom Zwischensockel abzukneifen. Bei Mißachtung können Sie mehrere Bausteine zerstören. Außerdem muß das Betriebssystem aus der 1541 ausgelesen und in ein 27128-EPROM gebrannt werden.

Der herausgelötete bzw. herausgehebelte 6522 ist im nächsten Schritt in den 40poligen Zwischensockel einzusetzen. Ist der 6522 eingelötet und nicht gesockelt, gilt das, was zu den Betriebssystemen gesagt wurde. Beim Einsetzen des Sockels achten Sie darauf, daß er in korrekter Richtung in der Fassung steckt. An der Seite der Kerbe (am Sockel) befindet sich Pin 1 und 40. Damit sind die erforderlichen Hardwareänderungen abgeschlossen, und das neue System »Hypra-Speed« kann getestet werden (Tabelle 3 zeigt noch einmal einen Überblick aller erforderlichen Teile.)

Bauen Sie alles vorsichtig wieder zusammen und schalten Sie Computer und Diskettenlaufwerk ein. Der Computer sollte sich jetzt mit einer leicht modifizierten Einschaltmeldung melden. Weigert er sich, sind alle Verbindungen nochmals zu überprüfen. Das System hat die Befehle:

Tabelle 3: Bauteileliste

Parallelkabel
1 Meter 10adriges Flachbandkabel
1 x User-Port-Stecker
1 x 40polige IC-Fassung mit langen Beinchen
1 x 40polige IC-Fassung normale Ausführung
Adapterplatine
1 x 28polige IC-Fassung normale Ausführung
2 x 14polige Stiftleisten
1 x 10 k+; 1/8 Watt
1 x Schalter (1 x um)
50 cm 2adriges Kabel



[4] Layout des 40poligen Zwischensockels für das Parallelkabel. Achtung: Das Layout ist spiegelverkehrt im Verhältnis 1:1.

SYS 0 oder SYS 0000 – stellt ein versehentlich mit NEW gelöschtes Basic-Programm wieder her, entspricht also dem in vielen Basic-Erweiterungen existierenden Befehl OLD oder RENEW.

SYS "- – schaltet den Speeder ab. Einige Programme arbeiten nicht mit Speedern (das sind in erster Linie solche mit eingebauten Softwarebeschleunigern). Für diesen Fall läßt sich »Hypra-Speed« mit dem Befehl »SYS"-« dauerhaft abschalten.

SYS" + – macht den Befehl »SYS"-« wieder rückgängig (schaltet »Hypra-Speed« wieder ein).

SYS" – schaltet die Geräteadresse des Diskettenlaufwerks von 8 auf 9 oder 9 auf 8 um.

SYS" – liest den Fehlerkanal der aktuellen Floppy aus.

SYS"$ – listet das Directory einer Diskette ohne Programmverlust.

LOAD"name",8,0 – lädt das Programm »name« an den Basic-Anfang.

Auch die Editierfunktionen wurden erweitert.

<CTRL -> – während des Listens eines Programms kann die Bildschirmdarstellung mit <CTRL> verlangsamt werden, mit <CTRL -> läßt sich das Scrollen anhalten.

<CTRL> und <£> – druckt eine Hardcopy vom Lowres-Bildschirm.

<CTRL INS/DEL> – löscht die aktuelle Bildschirmzeile.

<CTRL CLR/HOME> – setzt den Cursor in die untere Ecke des Bildschirms.

<CTRL RUN/STOP> – lädt das erste Programm von der Diskette und starte es mit RUN. Befindet sich das Directory auf dem Bildschirm, läßt sich ein Programm auch laden und starten. Sie setzen dafür den Cursor an den Anfang der Zeile des Programmnamens. Anschließend drücken Sie <SHIFT RUN/STOP>.

<SHIFT CBM> – entspricht <SHIFT RUN/STOP>

Wie bei jedem anderen Floppy-Speeder, wurden auch bei »Hypra-Speed« die Funktionstasten mit häufig benutzten Kommandos beziehungsweise Befehlen belegt.

<F1> – listet ein im Speicher stehendes Basic-Programm.

<F3> – startet ein im Speicher stehendes Basic.

<F5> – ähnelt der Tastenkombination <CTRL RUN/STOP>. Allerdings wird das Programm absolut, also an seine tatsächliche Adresse geladen und nicht gestartet.

<F7> – gibt das Directory ohne Programmverlust auf dem Bildschirm aus.

<F2> – gibt den Befehl SYS$C000 auf dem Bildschirm aus. Der Cursor steht dabei auf dem »C«. Dadurch läßt sich die Einsprungsadresse in ein zuvor geladenes Maschinenprogramm leicht ändern.

<F4> – schaltet zwischen Geräteadresse 8 und 9 um. Außerdem wird der Fehlerkanal gelesen und auf dem Bildschirm ausgegeben.

<F6> – gibt den String »SAVE"« auf dem Bildschirm aus.

<F8> – gibt den String »SYS"« auf dem Bildschirm aus. Mit diesem Befehl lassen sich Floppybefehle senden. So formatiert SYS"N:name,ID" eine neue Diskette.

Neben den bisher beschriebenen neuen Befehlen und Funktionen befinden sich in »Hypra-Speed« noch einige Leckerbissen:

35/40 Sektoren Betriebssystem

Mit unserem Beschleuniger lassen sich statt der herkömmlichen 35 jetzt auch 40 Sektoren formatieren. Dadurch steigert sich die Speicherkapazität einer Diskette von 664 auf 749 freie Blöcke. Zum Formatieren einer Diskette mit 40 Sektoren geben Sie ein:

```
SYS"D4
SYS"N:name,id"
```

Damit das Betriebssystem später erkennt, ob mit 40 Sektoren formatiert wurde, wird als Kennung der Wert $4a auf die Diskette geschrieben. Um eine Diskette mit 35 Sektoren zu formatieren, existiert in Hypra-Speed die Befehlsfolge

```
SYS"D3
SYS"N:name,id"
```

Eingebaute Centronics-Schnittstelle

Die Centronics-Schnittstelle ist immer am User-Port mit der Geräteadresse 4 aktiv, wenn kein serieller Drucker unter dieser Adresse angeschlossen ist. Bei der eingebauten Schnittstelle existiert keine Umsetzung eventuell vorhandener Sekundäradressen. Die Daten werden also ohne jegliche Konvertierung zum Drucker geschickt. Beachten Sie dies bei Grafikausdrucken. Hier wird bei den wenigsten Druckern ein brauchbares Ergebnis erzielt, wenn keine entsprechende Softwareumsetzung stattfindet.

Die RS232-Schnittstelle am User-Port ist dann aktiv, wenn kein Centronics-Drucker und kein Parallelbus angeschlossen ist. Außerdem ist mit dem Befehl SYS"- Hypra-Speed abzuschalten. Leider existiert für die fertigen Platinen und EPROMs keine Bezugsadresse. Wenden Sie sich daher, wenn Sie sich die Herstellung der Baugruppen nicht zutrauen, an einen fachkundigen Freund oder den Fachhandel. (gr)

Kurzinfo: Hyp.packed

Programmart: Explode-Programm für Install
Laden: LOAD"HY.PACKED",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: erzeugt die Files »DELTA.C64«, »DELTA.DOS«, und »CATCH!«
Benötigte Blocks: 27
Programmautor: Thomas Enders

Kurzinfo: Install

Programmart: Installationsprogramm für Hypra-Speed
Laden: LOAD"INSTALL",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: braucht auf Diskette die Files »DELTA.C64«, »DELTA.DOS«, und »CATCH!«
Benötigte Blocks: 8
Programmautor: Thomas Enders

Tabula Print

Tabula Rasa schwarz auf weiß

Bringen Sie mit diesem kurzen Zusatzprogramm der Tabellenkalkulation »Tabula Rasa« formatiertes Drucken bei. Viele Wahlmöglichkeiten lassen das Ergebnis endlich professionell erscheinen.

Sie erinnern sich sicher an »Tabula Rasa«, das Tabellenkalkulationsprogramm aus dem 64'er Sonderheft Nr. 68. Viele Anwender waren enttäuscht, da eine wichtige Funktion dem Programm bisher fehlte: Es konnte die Ergebnisse nicht drucken, sondern nur alle belegten Felder in der Reihenfolge des Anlegens. Leider war aus Speicherplatzgründen keine ordentliche Druckfunktion mehr möglich. Die Alternative dazu bietet ein kurzes Zusatzprogramm, das einen beliebigen Ausschnitt eines auf Diskette gespeicherten Arbeitsblattes druckt.

Soll ein Arbeitsblatt formatiert gedruckt werden, speichern Sie es von Tabula Rasa aus auf Diskette. Wählen Sie einen beliebigen Filenamen und verlassen Sie danach Tabula Rasa. Danach wird das Zusatzprogramm nachgeladen:

```
LOAD "TABULA PRINT",8
```

Gestartet wird mit RUN. Legen Sie jetzt die Diskette mit dem zu druckenden Arbeitsblatt ein. Der Filename wird eingegeben und mit <RETURN> bestätigt. Sollten darin Umlaute oder Sonderzeichen vorkommen, geben Sie statt dessen Fragezeichen ein, da außerhalb von Tabula Rasa keine deutschen Sonderzeichen erlaubt sind. »Zähneputzen« muß also »Z?hneputzen« heißen.

Nach Bestätigung mit <RETURN> erfragt das Tool, ob der Ausdruck einen linken Rand, z.B. zum Abheften, bekommen soll. Der kleinste Wert ist 1, dann wird kein Rand gedruckt. Ansonsten erzeugt der Computer bei Eingabe von n einen (n-1) Zeichen breiten linken Rand. Im Normalfall können Sie mit <RETURN> die Vorgabe 5 übernehmen. Geben Sie keinen Wert über 70 ein!

Danach wird der Ausschnitt des Arbeitsblatts bestimmt. Geben Sie hier als erstes die Nummer der obersten Zeile ein. Voreingestellt ist »1«. Danach ist die unterste Zeile des Ausschnitts fällig. Die Vorgabe lautet aus technischen Gründen 16. Ihnen sind hier nur folgende Grenzen gesetzt: Die unterste Zeile darf nicht kleiner als die oberste und nicht größer als 208 (mehr Zeilen hat ein Arbeitsblatt von Tabula Rasa nicht) sein. Ansonsten kann der Ausschnitt vertikal beliebig groß sein, nur genug Papier sollte vorhanden sein.

Nachdem diese Eingaben jeweils mit <RETURN> bestätigt wurden, erfolgen die letzten vier Eingaben nur durch Tastendruck. Als erstes ist die Startspalte des Ausschnitts zu bestimmen, irgendein Buchstabe von <A> bis <Z>. Analog dazu geben Sie bitte die Bezeichnung der letzten zu druckenden Spalte an.

Achtung: Das Programm schneidet, falls der Ausdruck zu breit wird, automatisch unter Berücksichtigung der eingestellten Schriftart (siehe unten) den rechten Rand ab. Sie brauchen sich also darum bei der Wahl des Ausschnitts keine Gedanken zu machen.

Auf vielen Druckern kann man Texte mit halber Schriftbreite drucken. Von Tabula Print aus können Sie jetzt mit <J> oder <N> diese Option ein- oder ausschalten.

Zuletzt fragt das Programm, ob in Wertfeldern die Formel oder das Ergebnis, also der numerische Wert gedruckt werden soll. Diese Umschaltung kennen Sie bereits vom Hauptprogramm Tabula Rasa. Antworten Sie mit <W>, wenn die errechneten Zahlen gedruckt werden sollen, oder mit <F>, wenn die Berechnungsformel gefragt ist.

Alle weiteren Angaben holt sich das Programm aus dem Arbeitsblatt, das nun nachgeladen wird. Findet das Programm nichts unter dem vorgegebenen Namen auf Diskette, erscheint eine entsprechende Fehlermeldung. Ebenso wird ein nicht betriebsbereiter Drucker erkannt und gemeldet.

Tabula Print druckt jetzt erst eine Überschrift, in der ganz rechts auch der im Arbeitsblatt enthaltene Titel erscheint; darunter der gewünschte Ausschnitt formatiert.

Tabula Print wurde für einen Fujitsu DL 900 Matrixdrucker in der Fujitsu-Emulation mit Wiesemann Interface 92000 geschrieben, funktioniert aber mit auch mit Epson-kompatiblen Druckern am seriellen Port. Der Betrieb mit den Druckern der MPS-Serie von Commodore ist nicht möglich.

Ist Ihr Drucker am Userport angeschlossen, laden Sie

```
LOAD"TABULA PRINT PAR",8
```

und starten ebenfalls mit RUN. Erweiterungen, die eine Userport-Centronics-Schnittstelle bedienen (z.B. Magic-Formel) sollten allerdings ausgeschaltet sein.

Werden alle Zeilen in eine gedruckt (also übereinander), schalten Sie Ihren Drucker in den Auto-Linefeed-Modus, da das Programm am Zeilenende nur ein CR, kein LF sendet.

Um Schmalschrift einzuschalten, wird der SI-Code (CHR$(15)) gesendet. Vorher schaltet das Programm in jedem Fall den Drucker mit ESC 2 auf einen Zeilenabstand von 1/6 Zoll. Weitere Druckerbefehle kommen nicht vor.

Der Drucker muß, wenn er auch die Umlaute und die eckigen Klammern korrekt wiedergeben soll, im IBM-Zeichensatz »page 437, Set II« arbeiten. Haben Sie den »german« Zeichensatz eingestellt, werden zwar die Umlaute richtig gedruckt, aber die eckigen Klammern nicht. Das Copyright-Symbol, das unter Tabula Rasa existiert, wird beim Ausdruck in den Kleinbuchstaben »c« gewandelt. (gr)

Kurzinfo: Tabula Print

Programmart: Drucker-Tool für Tabula Rasa
Laden: LOAD"TABULA PRINT",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 8
Programmautor: Nikolaus Heusler

Relativ einfach

Vielleicht ist die Verbindung zum ehrwürdigen Einstein für die unnötige Angst vor relativen Dateien verantwortlich. Sie werden sehen, daß sich gerade diese Dateiart einfach und schnell programmieren läßt, wenn Sie ein paar Kniffe beachten.

Als Commodore gegen Ende der siebziger Jahre den Vorgänger der Diskettenstation 1541 entwickelte, hatten Computer wie der PET 2001 in ihrer Grundausstattung nur 8 KByte Speicherkapazität. Trotzdem liefen auf diesen Rechnern sogar kommerzielle Dateiverwaltungen. Möglich wurde dies nur, weil Commodore einen Trick anwandte - die intelligente Diskettenstation. In diesen Floppies wurde damals schon ein eigenständiger Mikroprozessor (Steuerrechner) eingebaut, der auch ein eigenes Betriebssystem, speziell zur Verwaltung und Steuerung von Disketten besaß. Diese Linie hat Commodore auch bei den Floppies für den C 64 und C 128 nicht verlassen. Sogar das Betriebssystem ähnelt dem der alten Floppies.

Wir profitieren heute noch von diesem Konzept. Es bietet eine Entlastung des steuernden Computers. Er muß keine Programmfolgen übermitteln, was zu Speicherplatzproblemen führen würde, sondern nur Befehlswörter. Den Rest erledigt die Floppy automatisch.

Gerade bei Dateiverwaltungen bietet diese Arbeitsmethode wesentliche Vorteile. Eine Datei muß nicht komplett in den Speicher des Computers übernommen werden, um einen Datensatz zu verwenden. Sie läßt sich bei sequentiellen Dateien in der Reihenfolge des Schreibens Stück für Stück lesen. Beim Schreiben wird einfach der nächste Eintrag hinzugefügt. Da die Diskette mehr Daten faßt als der Computer, lassen sich damit auch aufwendige Aufgaben lösen. Leider hat die sequentielle Speicherung einen gewaltigen Nachteil: Wird ein Eintrag gesucht, muß die gesamte Datei danach durchsucht werden. Dadurch entstehen, abhängig von der Dateilänge, unerträgliche Wartezeiten. Zum Glück gibt es noch eine andere Art der Datenspeicherung - die relative Datei: suchen Sie hier beispielsweise den Eintrag (»Record«) Nummer 25, brauchen Sie sich nicht vorher durch die Records 1 bis 24 zu wühlen, so wie bei einer sequentiellen Datei (SEQ). Ein weiterer Vorteil ist, daß Sie keine Kopie der gesamten Datei machen müssen, nur um einen Datensatz zu ändern; es wird stattdessen einfach der alte Eintrag durch den neuen ersetzt - ganz unbürokratisch.

Das Prinzip von REL-Dateien

Jeder Eintrag belegt einen beim Anlegen der Datei vorgegebenen Platz und kann über diesen Platz direkt angewählt werden. Dadurch bildet ein gelöschter Datensatz zwar eine Lücke, die darauf folgenden müssen dafür aber nicht verschoben sein. Natürlich müssen auch nicht alle Sätze voll belegt werden. Sie könnten eine Datei mit einer Datensatzlänge von 60 erzeugen, obwohl einige (oder alle) Einträge weniger als je 60 Bytes belegen. Die restlichen Bytes werden mit Nullen aufgefüllt. Zusätzlich steht ein ständig aktualisierter Zeiger zur Verfügung - der Index. Bei jedem Zugriff führt das Betriebssystem der Floppy Berechnungen durch und vermerkt die Änderungen. Dadurch steht fest, wo die gesuchten Daten auf der Diskette liegen. Dazu existiert für jede REL-Datei ein »Side-Sector« (Zeigerblock), in dem die belegten Blocks verzeichnet sind. Die Zeigerblocks fungieren als Wegweiser durch das File. Dieser Vorgang läuft vom Programmierer unbemerkt im Hintergrund ab - zum Glück, denn er ist ziemlich kompliziert. Rufen wir beispielsweise den Datensatz 15 einer relativen Datei mit Datensatzlänge 100 ab, weiß das Laufwerk aufgrund seiner Berechnungen, daß es dazu beispielsweise den 6. Sektor der Datei lesen muß. Es sucht im Side-Sector die Adresse (Track, Sektor) und liest den Block. Danach ergeben weitere Berechnungen, daß sich der gesuchte Datensatz z.B. ab Byte 130 in diesem Block befindet. Auch dieser Vorgang läuft unbemerkt ab (wenn auch mit einigen kleinen Fehlern, wie wir später sehen werden). Sie geben nur die Datensatznummer an und lassen den Computer rechnen.

REL-Dateien in der Praxis

Es gibt aber Nachteile bei der Verwendung relativer Dateien. Sie sind größer und komplexer als SEQ-Files, und enthalten einige Fallen, in die man ohne Vorkenntnisse stürzt:

Betrachten wir dazu ein SEQ-File. Es ist einfacher als eine REL-Datei, und erledigt manche Aufgaben genauso gut. Eine Grundregel der Datenverarbeitung besagt: Falls Sie regelmäßig auf mehr als 15 Prozent eines Files zugreifen, sollten Sie eine sequentielle Datei verwenden. Die REL-Datei erlaubt zwar direkten Zugriff auf den Teil, den Sie suchen, muß aber zuerst die Sidesectoren lesen und berechnen. Eine sequentielle Datei dagegen wird vom Anfang bis zum Ende gelesen. Belegt eine sehr große sequentielle Datei sehr viel Platz auf

Diskette (über eine halbe Diskettenseite), ist auf jeden Fall die relative Datei vorzuziehen. Einen Nachteil der relativen Datei sollte man allerdings nicht verschweigen: Wird ein Datensatz geändert, geht der alte Inhalt unwiderruflich verloren. Ein Eingabefehler genügt da schon. Mit SEQ-Dateien erzeugen Sie die neue Version, indem Sie eine Kopie anlegen. Die alte Version ist, ggf. »unsichtbar«, immer noch vorhanden und könnte restauriert werden.

Das Demoprogramm

Den Aufbau einer REL-Datei sehen Sie am besten aus einem fertigen Programm. Es befindet sich mit auf Diskette und ist mit REM-Vermerken dokumentiert. Geladen wird es mit

```
LOAD"REL-DEMOPROGRAMM",8
```

Danach sollten Sie sich mit LIST einen Überblick verschaffen (Abb.). Im Textkasten finden Sie eine Zusammenstellung der wichtigsten verwendeten Diskettenbefehle.

Sieben Regeln

Sieben Grundregeln sollten Sie aus Gründen der Datensicherheit beachten. Sie hängen mit Betriebssystemfehlern des Laufwerk-DOS zusammen:

Regel 1: Erzeuge genügend Datensätze

Wenn Sie das erste Mal ein File anlegen, sorgen Sie dafür, daß mehr als ein Block auf Diskette von der Datei belegt wird. Das Beispielprogramm arbeitet etwa mit einer Datensatzlänge von 33, mithin sollten acht oder mehr Datensätze angelegt werden (254/33 = 8). Es werden zehn Datensätze erzeugt (Zeile 320). Eine REL-Datei kann später zwar erweitert werden, jedoch klappt das nicht immer reibungslos. Gönnen Sie daher der Datei lieber zu viele Einträge.

Beim Anlegen der Datei (Zeile 290) wird die Datensatzlänge auf 33 eingestellt. Ihnen fällt vielleicht auf, daß diese Angabe bei den OPEN-Befehlen in 190, 390 und 1400 fehlt. Anders als bei SEQ-Dateien (hier müssen Sie immer schreiben: OPEN 2,8,2,"name,S,R") ist diese Angabe beim Öffnen einer bereits bestehenden REL-Datei unnötig.

Wenn einmal ein File angelegt ist, muß es nur noch aktualisiert werden. Das Beispielprogramm prüft in den Zeilen 170 bis 240, ob die Datei schon angelegt ist. Falls nicht, übernehmen dies die Zeilen 280 bis 390. Das Programm akzeptiert als einzige Floppy-Fehlernummer die 62 (FILE NOT FOUND), alles andere ist ein »echter« Fehler, den die Diagnoseroutinen ab Zeile 1620 (vor dem Programmstop) ausgewerten.

Regel 2: Setze den Recordzeiger immer auf das erste Byte des Datensatzes

Der Zeiger im P-Befehl (s. Textkasten) sollte immer auf dem ersten Zeichen im Eintrag stehen. Der letzte Parameter des P-Kommandos sollte daher CHR$(1) sein (siehe Zeilen 670, 920, 1300). Lesen oder beschreiben Sie jedes Mal den ganzen Record in einem Aufwasch, so ersparen Sie sich Ärger.

Regel 3: Prüfe den Disk-Status

Lesen Sie immer den Status aus dem Fehlerkanal 15, nachdem Sie der relativen Datei einen Befehl gegeben haben. Sie brauchen den ermittelten Fehler überhaupt nicht weiter auszuwerten (Zeile 1360), durch das Lesen des Fehlerkanals gewinnt das Laufwerk die notwendige Zeit, um die gesuchte Stelle der Datei aufzuspüren und anzufahren.

```
260 REM ** DATEI ANLEGEN
270 :
280 PRINT:PRINT"DATEI WIRD ANGELEGT
290 OPEN1,8,2,"0:"+NN$+",L,"+CHR$(33):REM REL-FILE ANLEGEN, SATZLAENGE 33
300 INPUT#15,E,E$,E1,E2
310 IFE<>0THEN1620
320 PRINT#15,"P"+CHR$(98)+CHR$(10)+CHR$(0)+CHR$(1):REM SEK.ADR 98, DAT.SATZ 10
330 INPUT#15,E,E$,E1,E2
340 IFE<>50THEN1680
350 PRINT#1,CHR$(255);:REM DATEI FREIGEBEN
360 INPUT#15,E,E$,E1,E2
370 CLOSE1
380 IFE<>0THEN1620
390 OPEN1,8,2,"0:"+NN$
400 GOTO450
410 PRINT:PRINT"DATEI EXISTIERT SCHON!
420 :
430 REM ** HAUPTMENUE
440 :
450 PRINT:PRINT" [1] ALLE EINTRAEGE LESEN
460 PRINT" [2] EINEN EINTRAG LESEN
470 PRINT" [3] EINTRAG SCHREIBEN
480 PRINT" [4] EINTRAG LOESCHEN
490 PRINT" [5] DATEI LOESCHEN
500 PRINT" [6] SCHLUSS DAMIT!
510 PRINT:INPUT"1 BIS 6 ";X:PRINT
520 ONXGOTO570,810,870,1270,1470,1210
530 GOTO450
540 :
550 REM ** ALLE EINTRAEGE LESEN
560 :
570 R=0:O=0:REM O=0: FLAG ALLES LESEN
580 R=R+1:REM R=RECORD-NUMMER, F=FELDNUMMER
590 :
600 REM ** LESEROUTINE
610 :
620 F=0:X=0:H=INT(R/256):L=R-H*256:REM HIGH/LOWBYTE
630 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1):REM RECORD ANFAHREN
640 INPUT#15,E,E$,E1,E2
650 IFE=50ORO=2THEN450:REM RECORD BESTEHT NICHT
660 O=O*(O+1):IFE<>0THEN1620
670 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1):REM RECORD NOCHMAL ANFAHREN
680 INPUT#15,E,E$,E1,E2
690 IFE=50THEN450:REM RECORD BESTEHT NICHT
700 IFE<>0THEN1620
710 INPUT#1,X$:SW=ST:REM STATUS MERKEN
720 IFX$=CHR$(255)THENPRINT"EINTRAG"R"IST LEER":GOTO580
730 IFF=0THENPRINT"EINTRAG"R":"
740 F=F+1:IFSW=0THENX=1
750 PRINT"      FELD"F":"X$
760 IFSW=0THEN710
770 GOTO580
780 :
790 REM ** EINEN EINTRAG LESEN
800 :
810 INPUT"WELCHEN EINTRAG LESEN ";R
820 IFR<1ORR<>INT(R)THEN810
830 PRINT:O=1:GOTO620:REM FLAG O=1: NUR EINEN LESEN
840 :
850 REM ** EINTRAEGE SCHREIBEN
860 :
870 INPUT"BESCHREIBE EINTRAG NR.";R
880 IFR<1ORR<>INT(R)THEN870
890 H=INT(R/256):L=R-H*256:REM HIGH/LOWBYTE (EIGENTLICH UNNOETIG, DA R < 41)
900 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1)
910 INPUT#15,E,E$,E1,E2:IFE<>50ANDE<>0THEN1620
920 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1)
930 IFE=50THENPRINT:PRINT"DATENSATZ MUSS ERST GELOESCHT WERDEN!":GOTO
940 IFE<>0THEN1620
950 PRINT:INPUT"WIE VIELE FELDER (1-3) ";N
960 IFN<1ORN>3THEN950
970 :
980 REM ** FELDER EINGEBEN
990 :
1000 A$=""
1010 PRINT:PRINT"BITTE FELDDATEN EINGEBEN!
1020 FORJ=1TON
1030 IFN<>1THENPRINT"FELD"J;
1040 INPUTB$
1050 A$=A$+B$+CHR$(13):REM DATENSATZ ZUSAMMENFUEGEN
1060 NEXTJ
1070 A$=LEFT$(A$,LEN(A$)-1):REM LETZTES CR (CHR$(13)) ENTFERNEN
1080 PRINT#1,A$;:REM UND IN DATEI SCHREIBEN
1090 INPUT#15,E,E$,E1,E2
1100 IFE<>0THEN1620
1110 CLOSE1
1120 INPUT#15,E,E$,E1,E2
1130 IFE<>0THEN1620
1140 OPEN1,8,2,"0:"+NN$
1150 INPUT#15,E,E$,E1,E2
1160 IFE<>0THEN1620
1170 GOTO450
1180 :
1190 REM ** ENDE
1200 :
1210 CLOSE1
1220 CLOSE15
1230 END
1240 :
1250 REM ** EINTRAG LOESCHEN
1260 :
1270 INPUT"LOESCHE EINTRAG NR.";R
1280 IFR<1ORR<>INT(R)THEN1270
1290 H=INT(R/256):L=R-H*256:REM HIGH/LOWBYTE
1300 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1)
1310 INPUT#15,E,E$,E1,E2:IFE<>50ANDE<>0THEN1620
1320 PRINT#15,"P"+CHR$(98)+CHR$(L)+CHR$(H)+CHR$(1)
1330 IFE=50THENPRINT:PRINT"DATENSATZ NEU ANGELEGT!
1340 IFE<>0ANDE<>50THEN1620
1350 PRINT#1,CHR$(255);:REM LOESCHCODE IN DATEI SCHREIBEN
1360 INPUT#15,E,E$,E1,E2
1370 CLOSE1
1380 INPUT#15,E,E$,E1,E2
1390 IFE<>0THEN1620
1400 OPEN1,8,2,"0:"+NN$
1410 INPUT#15,E,E$,E1,E2
1420 IFE<>0THEN1620
1430 GOTO450
1440 :
1450 REM ** DATEI LOESCHEN
1460 :
1470 PRINT"DATEI WIRKLICH LOESCHEN ? [J/N]
1480 GETA$:IFA$="N"THEN450
1490 IFA$<>"J"THEN1480
1500 CLOSE1
1510 PRINT#15,"S0:"+NN$:REM SCRATCH-BEFEHL
1520 PRINT:PRINT"DATEI WURDE GELOESCHT.
1530 PRINT:PRINT" [1] PROGRAMM NEUSTART
1540 PRINT" [2] ENDE
1550 PRINT:INPUT"1 ODER 2 ";X:PRINT
1560 IFX=2THEN1210
1570 IFX=1THENRUN
1580 GOTO1550
1590 :
1600 REM ** DIAGNOSE-ROUTINE
1610 :
1620 PRINT:PRINT"VORSICHT: DISKETTENFEHLER!
1630 PRINT:PRINT"FEHLERNUMMER"E
1640 PRINT"TEXT: "E$
1650 PRINT"TRACK"E1" SEKTOR"E2
1660 CLOSE1
1670 STOP
1680 IFE<>0THEN1620
1690 PRINT:PRINT"ABARBEITUNG NICHT KORREKT!":GOTO1660
```

Das Demo-Listing zeigt alle Funktionen von REL-Dateien

Einige Fehlermeldungen treten im Normalbetrieb auf und zeigen den korrekten Ablauf an. So erwartet Zeile 340 einen Fehler Nr. 50. Der Grund: Vorher wurde die relative Datei angelegt, indem wir das »Freizeichen« CHR$(255) in den höchsten Datensatz (Nr. 10) schreiben. Dieser Satz existierte noch nicht, da die Datei ja eben erst angelegt wurde. Das Laufwerk läßt dies natürlich nicht ungestraft mit sich machen und reagiert etwas sauer, wenn auch zu Unrecht mit dem Fehler. Bei der Statusprüfung in Zeile 690 weist ein NO RECORD-»Fehler« darauf hin, daß es in der Datei keine weiteren Einträge mehr zu lesen gibt. Das Programm bricht die Schleife hier ab. Und in Zeile 930 informiert eben dieser Fehler den Anwender darüber, daß der momentan beschriebene Datensatz noch unberührt war.

Fehlerkanal lesen

Beachten Sie: Der Fehlerkanal muß ausgelesen werden. Dies dient nicht nur zur Fehlererkennung, sondern auch zum Bremsen des Programms. Sie stellen sicher, daß keine Daten zwischen Computer und Laufwerk verlorengehen, bevor die Floppy fertig ist.

Regel 4: Addiere 96 zur Sekundäradresse

Es ist nicht weiter schwer, beim P-Befehl (z.B. in Zeile 320, 670, 920 oder 1320) zur Sekundäradresse den Wert 96 zu addieren, damit das Drive die Datei richtig behandelt. Diese wurde in Zeile 190 mit der normalen Sekundäradresse 2 geöffnet, zu der wir dann in den genannten Zeilen 96 addieren (ergibt 98), um auf die Datei zuzugreifen.

Regel 5: Nur ein PRINT# pro Datensatz

Wenn Sie in ein File schreiben, sollten Sie den Befehl PRINT# pro Eintrag nur einmal benutzen und zugleich alle Felder des Datensatzes schreiben.

Betrachten Sie einmal die Zeilen 1000 bis 1080. Obwohl es in einem Datensatz verschiedene Felder gibt, werden sie in Zeile 1050 zusammengeschweißt, um sie in Zeile 1080 zusammen an die 1541 zu senden. Die einzelnen Felder werden mit CR (Carriage Return, CHR$(13), entspricht der <RETURN>-Taste) getrennt. Am Ende des Textes darf allerdings keins stehen. Daher wird dieses in Zeile 1070 ausgefiltert. Zum Lesen verschiedener Datenfelder gehen Sie vor, wie in den Zeilen 620 bis 770 beschrieben. Die System-Status-Variable wird hier verwendet um festzustellen, ob innerhalb des Datensatzes noch weitere Felder folgen.

Wenn Sie, wie in unserem Beispielprogramm, in sequentieller Reihenfolge auf die Einträge zugreifen (1, 2, 3, und so weiter), genügen diese Maßnahmen. Zum Lesen in willkürlicher Reihenfolge (Menüpunkt 2, ab Zeile 550) sind allerdings noch zwei weitere Merkpunkte zu beachten:

Beliebige Reihenfolge

Regel 6: Zum Lesen zweimal positionieren

Wenn Ihr Programm Datensätze in beliebiger Reihenfolge liest, geben Sie das Positionier-Kommando (P) zweimal hintereinander. Lesen Sie auch jedesmal den Status der Station. Um auf einen neuen Datensatz zu positionieren, liest das Laufwerk meist auch einen neuen Datenblock ein. Für die entsprechende Track/Sektorangabe benötigt es auf jeden Fall den entsprechenden side sector. Manchmal, wenn der neue Datenblock sehr weit vom alten entfernt liegt, sogar einen neuen. Erst danach holt das Betriebssystem die eigentlichen Datenblocks von der Magnetscheibe. Das alles kann viel Zeit kosten. Gönnen Sie daher dem Laufwerk eine Verschnaufpause und positionieren Sie zweimal. Das Demoprogramm bewerkstelligt dies im Bereich der Zeilen 630 bis 700.

Regel 7: Zum Schreiben der Datei erst einmal schließen und anschließend wieder öffnen

Befehle zu relativen Dateien:

1. Berechnung von Low- und High-Byte	
High-Byte berechnen	HB=INT(SN/256)
Low-Byte berechnen	LB=SN-HB*256
2. Anlegen einer REL-Datei	
Befehlskanal öffnen	OPEN 1,8,15
Datenkanal öffnen	OPEN 2,8,2,"name,L," +CHR$(SL)
Positionierung auf höchsten Datensatz	PRINT#1,"P"CHR$(98)CHR$(L-)CHR$(H)CHR$(1)
Freigeben des Satzes	PRINT#2,CHR$(255)
Kanäle schließen	CLOSE 2:CLOSE 1
3. Öffnen einer bestehenden REL-Datei	
Befehlskanal öffnen	OPEN 1,8,15
Datenkanal öffnen	OPEN 2,8,2,"name"
4. Lesen eines Datensatzes	
Positionierung auf den Datensatz	PRINT#1,"P"CHR$(98)CHR$(L)CHR$(H)CHR$(1)
Daten einlesen	INPUT#2,daten oder GET#2, daten
5. Schreiben eines Datensatzes (alle Daten in einem String!)	
Positionierung auf den Datensatz	PRINT#1,"P"CHR$(98)CHR$(L)CHR$(H)CHR$(1)
Daten schreiben	PRINT#2,daten
Kanäle schließen	CLOSE 2:CLOSE 1
Abkürzungen: SN = Satznummer, H = High-Byte, L = Low-Byte, SL = Satzlänge	
Der P-Befehl	PRINT#1,"P"CHR$(c)CHR$(l) CHR$(h)CHR$(p)
CHR$(c) übergibt die Sekundäradresse des OPEN-Befehles, unter dem die REL-Datei geöffnet wurde, plus 96. CHR$(l) und CHR$(h) übergibt im Format Low-Byte, High-Byte die Nummer des Datensatzes, auf den positioniert werden soll. CHR$(p) kennzeichnet die Stelle, auf die innerhalb des Eintrags positioniert werden soll. p sollte 1 sein.	

Schreibt das Programm Datensätze ohne Einhaltung der Reihenfolge, sollte das File zur Sicherheit nach jedem Schreibzugriff geschlossen und gleich wieder geöffnet werden (nach Auslesen des Status). Sie bezahlen dafür ca. eine Sekunde Programmlaufzeit, umgehen aber einen Fehler im Betriebssystem der Floppy und stellen sicher, daß keine Daten im Nirwana landen.

Wenn Records geschrieben werden, hat das Laufwerk eine Menge zu tun. Es führt nicht jedesmal tatsächlich auch Schreibzugriffe auf die Magnetschicht aus, wenn Sie den PRINT#-Befehl gehen, sondern speichert die Daten in einem Puffer und wartet, bis ein CLOSE-Befehl erfolgt oder mit dem P-Kommando ein anderer Eintrag angewählt wird. Die Diskettenstation müßte eigentlich diese geänderten Daten sicher behalten, bevor sie den nächsten Block liest. Leider ist das aber nicht immer der Fall. Manchmal kommt es vor, daß die 1541 schlicht und einfach vergißt, daß noch Änderungen ausstehen. Am besten schließen Sie das File und öffnen wieder. Genau das tut das Beispielprogramm in den Zeilen 1110 mit 1160 und 1370 mit 1420. Sicher, in diesem Demo wurde leicht übertrieben, da die Floppy genügend Zeit zur Speicherung hat, während der Anwender den nächsten Datensatz eingibt. Aber es sind Ihre Daten - Sicherheit geht vor.

Noch einige Fallen

Es gibt noch einige andere Anormalitäten, wenn Sie mit REL-Files zu tun haben. Die Fehler im DOS sind leider gravierender, als man glauben möchte. Bitte wundern Sie sich bei Ihren Experimenten nicht, wenn im Fehlerkanal plötzlich nie dagewesene Fehlermeldungen wie

```
49,,00,00
08,DISK,00,00
04,OPEN,00,00
06,NOT,00,00
```

stehen.

Auf Commodore-Floppies sollten relative Dateien nicht die ASCII-Zeichen 0 (Null-Byte) und 255 enthalten. Versuchen Sie, eines dieser Zeichen in einen Record einzubauen, werden Sie unangenehme Überraschungen erleben. Zeichen 0 wird verwendet, um den nicht belegten Teil des Eintrags aufzufüllen. Steht ein Null-Byte zwischen den Daten, kann die 1541 nicht mehr feststellen, wo der Datensatz endet und arbeitet falsch. Die 255 wird vom DOS verwendet, um einen leeren Datensatz zu markieren. Unser Beispielprogramm schreibt diesen Code in Zeile 350 beim Anlegen der Datei, in Zeile 1350, um einen Eintrag zu löschen und wertet ihn in 720 aus, um einen leeren Datensatz zu erkennen. Vorsicht daher bei diesem Code.

Relative Files mit Datensatzlängen 42, 58 oder 63 können von der 1541 nicht angelegt werden. Damit müssen Sie sich einfach abfinden. Grund: Aufgrund eines Fehlers im DOS wertet das System diese drei Zahlen als die ASCII-Codes für Stern, Doppelpunkt und Fragezeichen, und solche Zeichen sind für Disk-Befehle bekanntlich tabu.

Sie sollten REL-Dateien grundsätzlich nur auf leeren Disketten anlegen, da sie (wenn sie groß genug werden) die anderen Dateien auf der Diskette und schlimmstenfalls sogar das Directory ohne Rücksicht auf Verluste gnadenlos überschreiben. Auch das Erweitern einer solchen Datei (laut Handbuch »problemlos möglich«) ist daher nur mit Vorsicht zu genießen.

Sie verfügen jetzt über das notwendige Werkzeug, um mit relativen Dateien arbeiten zu können. Was noch fehlt, ist das Verständnis für die internen Abläufe:

REL-Dateien intern

Wie bereits erwähnt, besteht eine REL-Datei aus drei Teilen.

1. Der Eintrag im Directory.
2. Die Datenblocks, die wie bei einer SEQ-Datei verkettet sind.
3. Die Side-Sectors (Zeigerblocks), die als Wegweiser durch die Datei dienen.

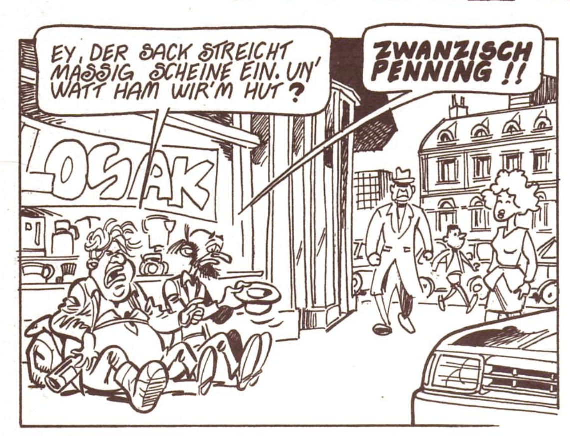
Jeder Side-Sector enthält in den ersten beiden Bytes einen Zeiger auf den nächsten Side-Sector. Byte 2 enthält die Nummer des Side-Sectors (0 bis 5), in Byte 3 steht die Datensatzlänge. Danach folgen in zwölf Bytes die Adressen (Track/Sektor) aller maximal sechs Side-Sector-Blocks einer REL-Datei (siehe unten). In den verbleibenden 256 - 2 - 2 - 12 = 240 Bytes stehen die Adressen von maximal 120 Datenblocks. Dadurch, daß in jedem Side-Sector die Adressen aller anderen Side-Sectors stehen, muß das Laufwerk nur einen Lesezugriff ausführen, um einen beliebigen Zeiger zu lesen (Geschwindigkeitsvorteil). Pro 120 angefangenen Datenblocks wird daher ein Side-Sector angelegt. Für eine neu formatierte Diskette, die 664 freie Blocks zur Verfügung stellt, ergibt sich demnach ein Bedarf von maximal sechs Zeigerblocks, dadurch bleiben bis zu 658 Blocks zur relativen Datenspeicherung übrig. Die kleinstmögliche REL-Datei besteht aus zwei Blocks (ein Side-Sector, ein Datenblock). Jeder Datenblock enthält 254 Datenbytes (die ersten beiden Bytes werden für die Verkettung benötigt). Damit stehen auf einer Diskette maximal 167132 Bytes für die relative Datenspeicherung zur Verfügung. Wir errechnen die maximal mögliche Anzahl von Einträgen bei gegebener Datensatzlänge, indem wir die 167132 durch die Satzlänge teilen. Auch hier gilt die theoretische Obergrenze von 65535 Sätzen (dies entspricht der höchsten mit zwei Bytes darstellbaren Zahl). Diese Länge kann nur bei einer Datensatzlänge von zwei Zeichen überschritten werden. Unsere Beispieldatei erlaubt bis zu 167132 / 33 = 5064 Einträge. Bei der höchstzulässigen Datensatzlänge von 254 Bytes erhalten Sie exakt 658 Datensätze.

Bei der Arbeit mit REL-Files muß der Programmierer daher einige Dinge besonders beachten, die auf den ersten Blick nicht logisch erscheinen. Aber wenn Sie die Regeln einhalten, werden Sie mit einer sehr flexiblen Art der Datenspeicherung belohnt, die nicht nur Programme professioneller macht, sondern auch große Vorteile in Komfort und Geschwindigkeit mit sich bringt. (gr)

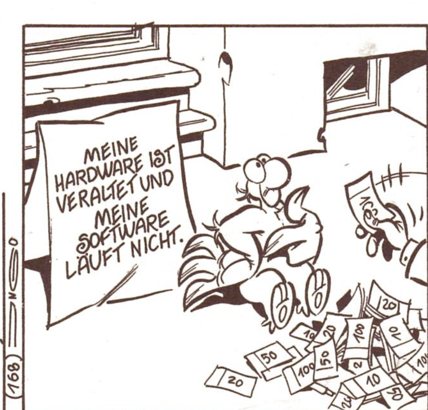
Literatur:
Immer der Reihe nach, 64'er 7/89, Seite 94
Kreuz und quer, 64'er 8/89, Seite 56
Kreuz und quer, 64'er 9/89, Seite 98
Floppy-Flops, 64'er 1/91, Seite 55
Floppy-Flops, 64'er 2/91, Seite 50
Karsten Schramm, Die Floppy 1541, Markt & Technik Verlag, Best.Nr. 90444
Lothar Englisch, Gerd Szczepanowski, Das große Floppy-Buch, Data Becker

Kurzinfo: REL-Demoprogramm

Programmart: Demo für REL-Dateien
Laden: LOAD"REL-DEMOPROGRAMM",8
Starten: nach dem Laden RUN eingeben
Besonderheiten: Dokumentiert den Aufbau einer relativen Datei (betrachten mit LIST)
Benötigte Blocks: 15
Programmautor: Nikolaus Heusler







ARC 1.5 - ein neues Tool für den C64

Der Superkleber

Aus der Welt der Personalcomputer sind Archivierungsprogramme nicht mehr wegzudenken. Nun gibt es endlich auch für den C64 ein Tool, um bis zu 45 Files zu verbinden.

Weder das Betriebssystem des C64 noch das Floppy-DOS erlauben die Kombination mehrerer Files zu einem. Genau das ist es, was ARC 1.5 macht.

Es entsteht ein einziges Programm, das später mit RUN gestartet wird. Danach werden die einzelnen Bestandteile wieder an ihre richtigen Positionen im Speicher geschoben. Vor allem auf PCs, aber auch beim Amiga, sind solche Tools, »Archive« genannt, Standard. Man verwendet sie z.B. zum Übertragen vieler kleiner Programme über Btx oder das Telefon (DFÜ). Hier ist es leichter (und billiger), ein kompaktes Gesamtpaket zu übertragen. Der Empfänger startet dieses dann, legt eine leere formatierte Diskette ein und die kleinen Files werden auf Diskette generiert.

Auch bei unserem Btx-Service verwenden wir dieses Verfahren mit dem ARC 1.5.

Ebenfalls zur Datensicherung sind diese Utilities geeignet. Stellen Sie sich vor, Sie haben so um die 40 kleine Files zu je etwa vier Blocks, die Sie beispielsweise auf Kassette überspielen wollen. Danach wandert die Kassette ins Archiv, um die Programme sicher zu verwahren.

Sie können nun alle 40 Programme nacheinander auf Band überspielen. Ein Packen jedes einzelnen Files ist jedoch nicht nur sehr arbeitsaufwendig, sondern auch nicht effizient, da hier der Entpacker schon so lang würde, daß der erzielte Platzgewinn wieder verloren wäre. Verbinden Sie aber mit einem »Archive«-Utility alle 40 Files zu einem Programm, wird es so um 162 Blocks lang (etwa zwei Blocks nimmt die kleine Routine in Anspruch, die nach dem Starten des Kompakt-Files die einzelnen Files wieder auf Diskette generiert). Durch späteres Packen läßt sich enorm viel Platz gewinnen.

»ARC« unterscheidet sich von herkömmlichen Linken dadurch, daß nach dem Start des Gesamt-Files die einzelnen Files wieder auf Disk erzeugt werden. Das in Maschinensprache geschriebene Tool kann dabei maximal 45 Programme verarbeiten. Sollen es noch mehr sein, müssen Sie die ersten 45 Files zusammenpacken, den Rest in ein zweites Gesamt-File speichern, um die beiden Kompakt-Files durch einen guten Packer zu schicken. Das Programm wird mit dem Befehl

```
LOAD "ARC 1,5",8
```

geladen und mit RUN gestartet. Nach dem Start sind zunächst die Namen der Einzel-Files einzugeben (Abb. 1). Die Eingabe ist jedesmal mit <RETURN> zu bestätigen. Sämtliche Eingaberoutinen des Programms sind so programmiert, daß durch Cursor-Bewegungen die Bildschirmmaske nicht zerstört werden kann. Vermeiden Sie bei der Eingabe Joker (»*« und »?«), um die File-Namen abzukürzen. Beim Laden gibt es zwar keine Probleme, da beim späteren Speichern jedoch wieder exakt dieselben Namen verwendet werden, sind Joker nicht erlaubt.

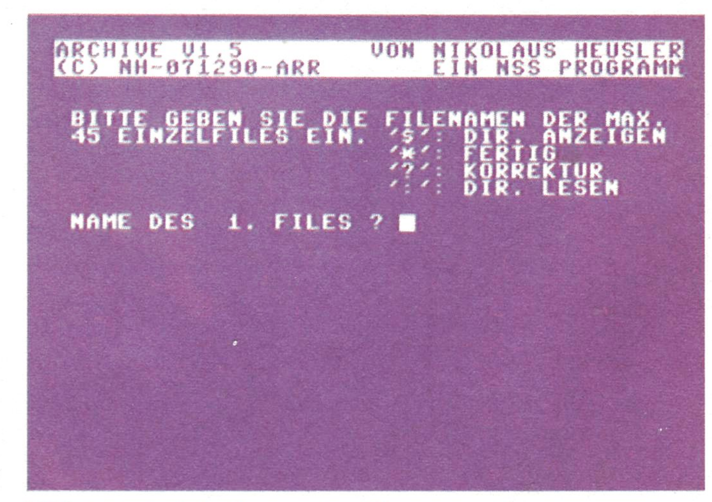
Durch Eingabe des Dollarzeichens an erster Stelle wird das Inhaltsverzeichnis der eingelegten Diskette (Directory) auf dem Bildschirm gelistet. Am Ende kann durch Tastendruck wieder in die Eingabeschleife zurückkehrt werden. Hinter dem Dollarzeichnen ist es möglich, das Directory genauer zu spezifizieren. So listet etwa »$HA*« alle Files auf, die mit »HA« beginnen (Abb 2).

Durch Eingabe eines Fragezeichens können Sie die letzte Eintragung korrigieren. Dies funktioniert natürlich nicht, wenn Sie erst bei der Eingabe des ersten Namens sind, oder der Cursor in der obersten Zeile steht. Die Liste der File-Namen wird bei der Eingabe gescollt.

Zum Beenden geben Sie einfach einen Stern ein. Dies funktioniert nur, wenn bereits mindestens ein File-Name festgelegt wurde.

Anstelle des 46. File-Namens nimmt das Programm nur noch das Dollarzeichen, das Fragezeichen und den Stern an, da mehr File-Namen nicht erlaubt sind.

Für den Fall, daß Sie sehr viele Routinen mit ähnlichen oder schwer zu merkenden Namen zusammenpacken wollen, ist es sinnvoll, die gewünschten File-Namen direkt aus dem Directroy zu übernehmen. Geben Sie dazu anstelle eines File-Namens einen Doppelpunkt ein und drücken <RETURN>. Die Bildschirmmaske wechselt und Sie werden aufgefordert, eine Diskette einzulegen - die Floppy mit den Quellprogrammen. Dann drücken Sie eine Taste. Mit <A> wird ARC 1.5 völlig neu gestartet (Datenverlust!). Jetzt zeigt das Tool den Diskettennamen an. Danach erscheinen der Reihe nach alle gespeicherten Files mit der dazugehörigen Länge in Blocks (254 Byte). Mit den Tasten <J> und <N> (für »ja« und »nein«) wählt man, ob das angezeigte File in das Gesamt-File übernommen werden soll. Auch in diesem Modus sind nur maximal 45 Files möglich, mehr faßt der Namensspeicher nicht. Das Programm zeigt laufend die Anzahl der bereits gewählten Files an. Wenn Sie mindestens ein File selektiert haben, beenden Sie die Auswahl, wenn Sie fertig sind, mit <*>. Sollen nur die ersten Files eines Directorys



[1] Zuerst sind die Namen der Files anzugeben

erfaßt und danach eine neue Diskette eingelegt werden, können Sie dies per <D> tun. Die Taste <A> dient zum Abbruch, hier wird ARC 1.5 neu gestartet, alle bisher erfaßten Namen werden vergessen.

Während der automatischen Namenserfassung aus dem Directory zeigt ARC 1.5 ständig die geschätzte Anzahl der belegten und die Anzahl der freien Blocks an. Dadurch wird vermieden, daß Sie später beim Einlesen die böse Überraschung - Speicher voll - erleben.

Der Speicherbereich für die Files erstreckt sich von $9CD bis $bFFF, das sind ca. 181 Blocks. Wird diese Anzahl überschritten, können mit <J> keine weiteren Files mehr selektiert werden. Die Länge der einzelnen Files wird übrigens dem Directory entnommen. Da diese grobe Abschätzung vor allem bei sehr kurzen Dateien (nur wenige Bytes) ungenau wird (Rundungsfehler), lassen sich auch bei angezeigtem Speicherüberlauf weitere Files mit <SHIFT J> übernehmen. Natürlich auf die Gefahr hin, daß später beim Einlesen »Speicher voll« erscheint.

Beispiel: Der Zähler zeigt 166 belegte und 15 freie Blocks (Summe 181). Jetzt können mit <J> nur noch Files mit 15, 14, 13 usw. Blocks Länge aufgenommen werden, keine Files über 15 Blocks Länge. Diese lassen sich nur mit <SHIFT J> erfassen. Im Normalfall sollten Sie die Abschätzung allerdings als obere Grenze akzeptieren, mit einer weiteren Einschränkung: In diese Berechnung gehen keine Files ein, die im ersten Modus (manuelle Eingabe der Filenamen) erfaßt wurden. Haben Sie den Automatikmodus erst aktiviert, nachdem schon einige File von Hand eingegeben wurden, werden diese nicht mitgerechnet, der Speicher kann dann u.U. sogar weniger als 181 Blocks fassen. Es ist nicht, auch nicht mit <SHIFT J>, möglich, 45 Files zu überschreiten.

Manuell oder Automatic

Vom Modus des automatischen Einlesen können Sie nicht in den Modus der manuellen Eingabe zurückkehren, wohl aber umgekehrt. Soll das Gesamt-File also Files enthalten, deren Namen Sie manuell über Tastatur eingeben wollen (Modus 1) und weitere, deren Namen Sie einlesen lassen wollen (Modus 2), so sind zuerst in Modus 1 die manuellen Dateinamen einzugeben, danach ist durch Eingabe des Doppelpunkts Modus 2 zu aktivieren und die restlichen Namen sind zu spezifieren. Danach kann mit der Stern-Taste der Vorgang beendet werden.

Bitte beachten Sie auch, daß es im Modus 2 u.U. Probleme bei Directory-Manipulationen geben kann, etwa dann, wenn nach dem zweiten Anführungszeichen noch Ladehilfen wie »8,1« stehen, oder wenn der Filename ein Directory-Cursor-Steuerzeichen oder Joker (»*« oder »?«) enthält. Insbesondere könnten Schwierigkeiten auftreten, wenn die im Directory verzeichnete Blocklänge nicht mit der tatsächlichen File-Länge übereinstimmt, da dann die oben erklärte Abschätzung falsch arbeitet, oder wenn Files auf der Diskette vorhanden sind, deren Länge mehr als 255 Blocks beträgt (kommt im Normalfall nicht vor, solch lange Files lassen sich mit ARC gar nicht bearbeiten). Wenden Sie Modus 2 nur bei Disketten an, die nicht z.B. mit dem »Disc-Wizard« manipuliert wurden!

Nach der Eingabe aller Namen ist zu wählen, ob alle Programme auf derselben Diskette stehen, oder ob nach jedem File auf einen Tastendruck gewartet werden soll, um Zeit für den Diskettenwechsel zu haben. Antworten Sie hier einfach mit <J> oder <N>.

Die folgende Frage erwartet den File-Namen, unter dem das gelinkte Gesamt-File gespeichert werden soll. Hier führt die Eingabe eines Sterns oder eines Fragezeichens zurück zur Frage, ob alle Files auf einer Diskette gespeichert sind. Vermeiden Sie auch hier die Eingabe verbotener Zeichen wie »*«, »?«, »=«, »:«, »$« usw.





[2] Bei der Directory-Anzeige ($) lassen sich auch Wildcats verwenden

Danach liest ARC alle benötigten Files in den Speicher. Diskettenfehler (z.B. »FILE NOT FOUND«) werden angezeigt. Der Anwender kann dann durch die Taste <V> das File (eventuell von einer anderen Diskette) erneut laden, mit <N> dieses File »vergessen« und mit dem nächsten fortfahren, oder mit <E> den Abbruch des Einlesevorgangs veranlassen. Dann wird das Gesamt-File gespeichert.

Den eingelesenen Programmen steht insgesamt ein Speicherplatz von etwa 45 KByte zur Verfügung, dennoch kann es vorkommen, daß ein File zu lang ist. Auch dieser Fall wird abgefangen. Es kann gewählt werden, ob das File zu überspringen oder der Einlesevorgang abzubrechen ist. Nach dem Einlesen wird das Gesamt-File auf Diskette gespeichert. Hier werden Sie aufgefordert, auf alle Fälle eine neue Diskette einzulegen, diese Aufforderung muß wie üblich mit einem Tastendruck bestätigt werden. Tritt ein Fehler beim Speichern auf, kann gewählt werden, ob ARC aufgeben soll oder ob die Speicherung zu wiederholen ist. Am Ende erscheint dann noch die Frage, ob Sie ARC noch einmal starten wollen (Taste <J>), oder ob ein Reset ausgelöst werden soll (<N>).

Das erzeugte Gesamt-File, das sich natürlich auch problemlos etwa auf Kassette speichern läßt, können Sie ganz normal wie ein Basic-Programm laden und nach Einlegen einer Diskette mit genügend freiem Platz per RUN starten. Jetzt werden automatisch die Einzel-Files mit ihrem Originalnamen wieder erzeugt. Das Programm zeigt dabei die Nummer des gerade generierten Files an.

Es bleibt nur noch zu sagen, daß sowohl das Generatorprogramm ARC als auch die Routine im Gesamt-File die zuletzt aktive Diskettenstation ansprechen. Dazu wird die aktuelle Geräteadresse geprüft. Liegt sie nicht zwischen 8 und 15, so wird sie automatisch auf 8 gesetzt. Durch dieses Verfahren ist sichergestellt, daß nicht aus Versehen ein Drucker, Plotter o.ä. angesprochen wird.

Wir wünschen Ihnen viel Erfolg beim Zusammenfügen Ihrer Programme! (gr)

Kurzinfo: ARC 1.5

Programmart: Archivierung
Laden: LOAD"ARC 1.5",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 14
Programmautor: Nikolaus Heusler

Floppy-zauber

Endlich Schluß mit dem umständlichen Floppykauderwelsch: Kompaktbefehle ersetzen ganze Programme

Die Diskettenlaufwerke für den C 64 unterscheiden sich in einem wesentlichen Punkt von denen anderer Computer: Sie verfügen über einen eigenen Computer. Leider werden die Vorteile dieser »intelligenten« Laufwerke durch ein unkomfortables, auf Kassettenbetrieb ausgelegtes Basic zunichte gemacht. Viele verwenden darum Floppy-Speeder, die durch ein geändertes Betriebssystem bereits einen recht komfortablen Umgang mit der Diskettenstation gestatten und dazu die Diskettenoperationen erheblich beschleunigen.

Disc-Basic hingegen bietet eine Reihe mächtiger Befehle und Funktionen für die vorher ganze Programme notwendig waren:
- 33 neue Befehle
- 10 neue Funktionen
- Labelverarbeitung
- Editierhilfen
- neuer Dateityp RECORD
- 6 neue Befehle und 4 Funktionen für relative Dateiverwaltung
- allgemeine Funktionen
- erweiterte Fehlerbehandlung

Sie laden von der beiliegenden Diskette mit

```
LOAD"DISC-BASIC",8
```

und starten nach dem Ladevorgang mit RUN. Anschließend meldet sich die Erweiterung mit einem neuen Einschaltbild. Ab jetzt stehen Ihnen die Befehlserweiterungen zur Verfügung:

Editorfunktionen

Der Editor ist mit einem bidirektionalen Scrolling ausgestattet und läßt sowohl zeilenweises als auch seitenweises Bewegen durch ein Basic-Listing, sowohl nach oben als auch nach unten zu.

<**CBM Cursor abwärts**> - zeilenweise nach unten
<**CBM Cursor rechts**> - zeilenweise nach oben
<**CTRL Cursor abwärts**> - seitenweise nach unten
<**CTRL Cursor rechts**> - seitenweise nach oben

Labelverarbeitung

Beim Basic des C64 sind Ziele von Sprungbefehlen immer Zeilennummern (z.B. GOTO100, GOSUB110). Bei einem Label hingegen wird das Sprungziel mit einer Zeichenkette bezeichnet. Vorteil dieser Methode ist eine übersichtlichere Bezeichnung der einzelnen Programmteile. In Disc-Basic wird diese Art von Programmteilen immer mit »LABEL« eingeleitet.

Achtung: Das Schlüsselwort LABEL muß grundsätzlich das erste Wort in einer Programmzeile sein. Es darf nicht als Zweitanweisung (durch einen Doppelpunkt getrennt) in einer Zeile stehen.

Verwenden Sie keine Basic-Wörter (END, THEN usw.). Diese werden nach der Eingabe in Token umgewandelt und daher nicht mehr gefunden (UNDEF'D STATEMENT ERROR).

Drei Befehle erleichtern den Umgang mit Labels. Dabei lassen sich diese auch als Variable verwenden:

JUMP - Springe zum Programmteil

... entspricht weitestgehend dem Basic-Befehl GOTO, verzweigt aber zu einem Programmteil, der mit LABEL eingeleitet wird:

```
10 LABEL ENDLOS
20 PRINT"SCHLEIFE"
30 JUMP ENDLOS
```

In Zeile 10 wird das Label ENDLOS deklariert. Zeile 20 gibt das Wort »SCHLEIFE« am Bildschirm aus. Zeile 30 verzweigt zurück zu »ENDLOS« (Zeile 10). Dieser Vorgang wird so lange wiederholt, bis Sie mit <RUN/STOP> unterbrechen. Für »ENDLOS« kann auch eine Variable Verwendung finden. Die Routine funktioniert auch:

```
5 A$ = "ENDLOS"
10 LABEL ENDLOS
20 PRINT"SCHLEIFE"
30 JUMP A$
```

Damit ist eine Beeinflussung des Sprungziels möglich.

SUB - Springe zur Unterroutine

... entspricht dem Befehl GOSUB. Im Unterschied dazu wird ein LABEL angesprungen. Dieses muß mit RETURN beendet werden.

```
10 SUB TEST
20 END
30 :
100 LABEL TEST
110 PRINT"PROGRAMMENDE"
120 RETURN
```

In Zeile 10 wird die Unterroutine »TEST« (ab Zeile 100) aufgerufen und der Text »PROGRAMMENDE« am Bildschirm ausgegeben. Danach verzweigt RETURN zurück zu Zeile 20 und das Programm wird beendet. Auch bei diesem Befehl lassen sich die Labels über Variable aufrufen.

SHOW - zeige Label

... listet ohne weitere Angaben alle im Programm verwendeten Labels auf:

```
SHOW TEST
```

listet das Programm ab dem Label Test.

Editierhilfen

Neun neue Anweisungen und Befehle stehen für die Programmierung zur Verfügung. Sie erfüllen alle Wünsche vom automatischen Vor- oder Umnumerieren der Programmzeilen bis zum Verschieben ganzer Programmblöcke.

AUTO xx,i - automatische Zeilenvorgabe
xx (Startzeile)
i (Schrittweite 0-255)

Beendet wird die automatische Zeilennumerierung durch eine Leereingabe (nur <RETURN>). Um an dieser Stelle wieder fortzufahren, genügt die Eingabe von AUTO ohne Parameter.

DELETE xxyy - löschen von Programmabschnitten
xx (Startzeile)
yy (Endzeile)

DELETE löscht den Programmblock von xxyy. Beispiele:

DELETE 100-200 löscht die Zeilen von 100 bis 200 einschließlich; DELETE 100- löscht die Zeilen ab 100 bis zum Programmende; DELETE -200 löscht die Zeilen vom Programmanfang bis 200 einschließlich.

GATHER xxyy,zz,i - Schieben von Programmblöcken
xx (Startzeile)
yy (Endzeile)
zz (neue Startzeile)
i (Schrittweite)

GATHER dient zum Verschieben von Programmblöcken. Beispiele: GATHER 100-200,500,1 verschiebt die Zeilen von 100 bis 200 einschließlich nach 500 und numeriert diesen Programmblock mit einer Schrittweite von eins neu durch. Die Eingabe der Schrittweite kann auch entfallen, dann wird ein Inkrement von 10 angenommen. Zeilenreferenzen bei GOTO oder GOSUB werden nicht mit geändert. Dies ist jedoch nicht von Bedeutung, da Disc-Basic mit der Labelverarbeitung Zeilennummern weitgehend überflüssig macht. Um Programmblöcke möglichst schnell verschieben zu können, benutzt GATHER das RAM unter dem E/A-Bereich als Zwischenspeicher. Da dieses RAM nur 4 KByte umfaßt, tritt bei Programmblöcken, die diese Länge überschreiten, die Fehlermeldung BLOCK TOO LARGE ERROR auf.

OLD - Restaurieren eines gelöschten Files

... holt ein mit NEW, Soft- oder Hardware-Reset gelöschtes Programm zurück.

RENUM xx,yy,i - umnumerieren von Basic-Zeilen
xx (Startzeile)
yy (neue Startzeile)
i (Schrittweite)

RENUM 10,100,10 numeriert ein Programm beginnend bei Zeile 10 mit neuer Startzeile 100 und Inkrement 10 neu durch. Die Angabe der Schrittweite kann enfallen, dann wird ein Inkrement von 10 angenommen. Beispiel:

Vorher:

```
5  REM -----------
10 REM -         -
11 REM - Beispiel -
12 REM -         -
90 REM -----------
```

Nach RENUM 10,100,10

```
5   REM -----------
100 REM -         -
110 REM - Beispiel -
120 REM -         -
130 REM -----------
```

Auch hier werden Zeilenreferenzen (GOTO, GOSUB etc.) nicht mitgeändert.

KEY x,"text" - Belegen der Funktionstasten
x (Nummer der Funktionstaste 1-16)
"text" (beliebiger Text mit max. 16 Zeichen)

KEY dient zur Belegung der Funktionstasten. Besonderheiten: RETURN kann durch »_«, das Hochkomma kann durch »'« eingegeben werden. Beispiele:

```
KEY 1,"LIST_" :REM belegt F1 mit LIST+RETURN
KEY 2,"LOAD '$',8" :REM belegt F3 mit LOAD "$",8
```

Es stehen 16 Funktionstasten zur Verfügung:

```
1=<F1> 5=<SHIFT>+<F1> 9=<CBM>+<F1> 13=<CTRL>+<F1>
2=<F3> 6=<SHIFT>+<F3> 10=<CBM>+<F3> 14=<CTRL>+<F3>
3=<F5> 7=<SHIFT>+<F5> 11=<CBM>+<F5> 15=<CTRL>+<F5>
4=<F7> 8=<SHIFT>+<F7> 12=<CBM>+<F7> 16=<CTRL>+<F7>
```

SET – Funktionstasten einschalten

Die Funktionstastenbelegung wird mit SET aktiviert. Nach <RUN/STOP> + <RESTORE> muß wieder eingeschaltet werden.

OFF – Funktionstasten ausschalten

... schaltet vorher belegte Funktionstasten ab.

DISPLAY – Anzeige der Funktionstasten

... gibt die Belegung am Bildschirm aus.

Datentyp

Disc-Basic stellt einen neuen Datentyp zur Verfügung: RECORD. Syntax: RECORD .ab = a1$,a2$,...,an$. ab ist die Recordvariable. Mit RECORD werden mehrere Strings zu einem Verbund zusammengefaßt. Der RECORD-Datentyp wurde ausschließlich für die Befehle RELREAD und RELWRITE geschaffen. Seine Verwendung ist dort beschrieben.

Befehle für relative Dateien

Es ist zwar möglich, mit dem Standard-Basic V2.0 relative Dateien zu verwalten – aber nur äußerst unkomfortabel und umständlich. Disc-Basic bietet jetzt sechs leistungsfähige Befehle und vier praktische Funktionen:

RELNEW (Gn) "Name",Anzahl,Länge – legt eine neue relative Datei an

Gn (Gerätenummer (normalerweise 8))
"Name" (Name der relativen Datei)
Anzahl (Anzahl der Datensätze)
Länge (Länge der Datensätze)

Die Angabe der Gerätenummer (Gn) kann auch entfallen, dann wird auf das Laufwerk mit der Gerätenummer 8 zugegriffen. RELNEW legt eine relative Datei mit dem Namen »Name«, der Datensatzanzahl »Anzahl« und der Datensatzlänge »Länge« an.

Achtung: Wenn eine Datei mit RELNEW angelegt wurde, ist sie geschlossen.

RELOPEN (Gn) "Name" – Öffnen einer relativen Datei

Gn (Gerätenummer, kann wie bei RELNEW entfallen)

Beim Öffnen einer relativen Datei mit RELOPEN wird der Dateizeiger auf den ersten Datensatz gesetzt.

Achtung: Es kann immer nur eine relative Datei geöffnet sein. Es ist allerdings möglich, daneben noch eine sequentielle Datei zu öffnen.

RELCLOSE – Schließen einer aktiven, relativen Datei

RELWRITE A$;"x" / RELWRITE .ab – schreiben in relative Datei

A$ (Stringvariable)
"x" (beliebiges Zeichen, auch als CHR$(Zahl))

Mit RELWRITE ist es möglich, einen String oder einen ganzen Record in eine geöffnete relative Datei zu schreiben. Beispiele: RELWRITE "Testtext" schreibt die Zeichenkette »Testtext« und ein CHR$(13) (RETURN) in den Datensatz, auf den der Dateizeiger zeigt. Jetzt zeigt der Dateizeiger auf das nächste Element. RELWRITE A$ entspricht dem vorhergehenden Beispiel, jedoch wird hier eine Stringvariable verwendet. Der String A$ wird in den Datensatz geschrieben, der Dateizeiger weist hinter den letzten Buchstaben. RELWRITE A$;"." schreibt A$ in den Datensatz, zusätzlich wird noch ».« angehängt, der Dateizeiger zeigt hinter den Punkt.

RELREAD A$;"x" / RELREAD .ab – Lesen aus relativer Datei

Mit RELREAD wird ein String oder ein Record eingelesen. Beispiele: RELREAD A$ liest den Datensatz, auf den der Dateizeiger weist, bis zum RETURN aus und weist die Zeichenkette dem String A$ zu. Der Dateizeiger wird jetzt eine Position weiter gesetzt. RELREAD A$;"." wie Beispiel zuvor, nur ist hier ».« das Abbruchkriterium. RELREAD .ab liest einen Record ein und setzt den Dateizeiger weiter. Es ist zu beachten, daß die Stringanzahl von geschriebenem und zu lesendem Record übereinstimmt.

SEEK Nummer, Stelle – Positionieren des Dateizeigers

Nummer (Nummer des Datensatzes)
Stelle (Position im Datensatz)

SEEK positioniert den Dateizeiger auf die angegebene Position. Beispielsweise setzt SEEK 20,4 den Zeiger auf Datensatz 20 und Stelle 4. Die Angabe der Stelle kann entfallen: SEEK 20,1 entspricht SEEK 20. Wird der Dateizeiger über das Ende der Datei positioniert, wird die Datei automatisch erweitert.

Funktionen für relative Dateien

FILESIZE – Anzahl der Datensätze

Beispiele: PRINT FILESIZE gibt die Anzahl der Datensätze aus; A=FILESIZE legt die Dateilänge in der Variablen a ab.

COMPSIZE – Datensatzlänge

Beispiel: PRINT COMPSIZE gibt die Datensatzlänge aus.

FILEPOS – Datensatznummer

... enthält die aktuelle Datensatznummer, auf die der Dateizeiger positioniert ist.

COMPPOS – Datensatzposition

... enthält die Position innerhalb des Datensatzes.

Wenn keine relative Datei geöffnet ist, ergeben die Funktionen FILESIZE, COMPSIZE, FILEPOS, COMPPOS den Wert null.

Allgemeine Diskettenbefehle

... dienen in erster Linie zur komfortablen Verwaltung der Floppy. Umständliche Open-Befehle sind nicht mehr nötig.

DISC (Gn) "x:text" – Floppybefehle senden

Gn (Gerätenummer)
x (Befehl (I,V,N,S,..))

... sendet einen Befehl zum Laufwerk. Beispiel:

```
DISC "V"
```

führt den Validate-Befehl aus. Die Angabe der Gerätenummer Gn kann entfallen, Default-Wert ist 8 (DISC "I" entspricht DISC(8) "I"). Dies gilt für sämtliche nun folgenden Befehle.

DIR (Gn) – Directory

Das Inhaltsverzeichnis der Diskette wird auf dem Bildschirm ausgegeben. Ein im Speicher befindliches Basic-Programm bleibt erhalten. Die Ausgabe kann mit der SHIFT-Taste angehalten und mit der RUN/STOP-Taste unterbrochen werden.

BLOAD (Gn) "Name" / BLOAD (Gn) "Name",Adresse – absolutes Laden

... lädt das Programm mit dem Namen »Name« an die angegebene Adresse. Ist keine Adresse angegeben, so wird absolut geladen: BLOAD "Test" entspricht LOAD "Test",8,1.

BLOAD "Test",49152 lädt das Programm »Test« nach 49152, ohne die Basic-Zeiger zu beeinflussen. Das sonst notwendige NEW zum Richten der Zeiger entfällt.

BSAVE (Gn) "Name",Startadresse,Endadresse – absolutes Speichern

... speichert den Speicherbereich von Startadresse bis Endadresse (ausschließlich) unter dem angegebenen Namen auf Diskette. Beispiel:

```
BSAVE "Test",49152,53248
```

speichert den Bereich von 49152 bis 53248 unter dem Namen »Test«.

MODULE (Gn) "Name",xxyy – Speichern von Programmteilen

xx (Startzeile)

yy (Endzeile)

MODULE "Test",100-200 speichert die Zeilen 100 bis 200 (einschließlich) unter dem Namen »Test«.

MERGE (Gn) "Name",xx – Zusammenfügen von Programmen

x (Startzeile)

MERGE lädt das Programm »Name« und hängt es an das im Speicher befindliche Programm an. Der neue Programmteil wird mit xx beginnend in Zehnerschritten neu durchnumeriert. Beispiel: Im Speicher befindet sich folgendes »Programm«:

```
10 PRINT "Das ist"
20 PRINT "Teil 1"
```

Auf Diskette ist das Programm »Teil 2« gespeichert, welches folgendermaßen aussieht:

Record-Handhabung:

Deklarierung eines Records:

```
10 RECORD .ad = NM$,ST$,OT$
```

Die Strings NM$, ST$, OT$ können wie gewohnt gehandhabt werden:

```
20 NM$="Gerald Kühne":ST$="Am Bängertchen 17a":OT$="
6798 Kusel"
```

Schreiben eines Records:

```
30 RELWRITE .ad
```

Die im Record deklarierten Strings NM$, ST$, OT$ werden durch ein Trennzeichen (CHR$(160)) getrennt in den Datensatz geschrieben, auf den der Dateizeiger zeigt.

Allgemeine Hinweise:

Verwenden Sie keine Soft- oder Hardware-Floppy-Speeder! Verwenden Sie keine geänderten Betriebssysteme! (Sie können Disc-Basic mit einem Floppyspeeder oder einem geänderten Betriebssystem laden und nach Abschalten des Speeders oder Umschalten auf das Originalbetriebssystem, gefolgt von Reset, mit SYS 2075 starten). Die Datasetten- und RS232-Routinen sind außer Funktion, wenn Disc-Basic aktiviert ist.

```
10 PRINT "Und das"
20 PRINT "ist Teil 2"
```

Nach MERGE »Teil 2«,10000 steht folgendes im Speicher:

```
10 PRINT "Das ist"
20 PRINT "Teil 1"
10000 PRINT "Und das"
10010 PRINT "ist Teil 2"
```

DNEW (Gn) "Name,ID" – Formatieren einer Diskette

Name (Diskettenname)

... dient zum schnellen Formatieren von Disketten. Die Angabe der ID ist verbindlich, sonst gibt das Laufwerk einen Syntax-Fehler aus. Beispiel: DNEW "Test,01" formatiert die im Laufwerk befindliche Diskette mit dem Namen »Test« und der ID 01.

DSCPOKE (Gn) Adresse,Byte – POKEn in Floppyspeicher

... entspricht dem POKE-Befehl im Standard-Basic, nur wird hier in den Floppyspeicher geschrieben. Beispiele: DSCPOKE 68,10 schreibt in die Floppyadresse 68 den Wert 10. Damit lassen sich Daten manipulieren.

```
DSCPOKE (8) 119,$20+9:DSCPOKE (9) 120,$40+9
```

ändert die Geräteadresse des Laufwerks 8 auf 9.

DSCSYS (Gn) Adresse – Start von Floppymaschinenprogramm

... startet ein Maschinenspracheprogramm an der angegebenen Adresse im Floppyspeicher. Beispiel: DSCSYS 49608 führt zum Fehlerblinken der Diode.

BLREAD (Gn) Track,Sektor,Adresse – Lesen eines Floppy-Blocks

... liest den Block, dessen Position durch Track und Sektor angegeben ist, an die angegebene Adresse. Es können auch Parameter entfallen. Beispiele: BLREAD 18,1,4096 liest den Diskettenblock 18,1 an die Adresse 4096. BLREAD liest den Folgeblock (sofern vorhanden) an die vorher verwendete Adresse. In diesem Fall werden Track und Sektor des Folgeblocks aus den Adressen 4096+4097 geholt und der Folgeblock an die Adresse 4096 geladen. BLREAD 12,9 lädt den Block 12,9 an die vorher verwendete Adresse.

Unter Disc-Basic ist ein spezieller Speicher für einen Diskettenblock vorgesehen, er liegt bei 61371 ($EEFB). Wird BLREAD zum ersten Mal verwendet, so wird, wenn keine Adresse angegeben ist, der Block nach 61371 geladen.

BLWRITE (Gn) Track,Sektor,Adresse – Schreiben eines Floppyblocks

... hier wird der Block (256 Bytes), der ab »Adresse« im Computerspeicher liegt, an die Position, die in »Track« und »Sektor« angegeben ist, gespeichert. Beispiele: BLWRITE 18,1,4096 speichert die Bytes von 4096 bis 4096+255 als Block 18,1 ab. BLWRITE speichert den zuletzt gelesen Block an die zuletzt verwendete Position. BLWRITE 18,1 speichert den Block aus der zuletzt benutzten Adresse als Block 18,1 ab.

Nehmen wir an, Disc-Basic wurde gerade gestartet, die Default-Einstellung für den Blockspeicher ist 61371 ($EEFB).

```
10 BLREAD 18,1: REM Block 18,1 nach 61371
20 POKE 61376,255: REM Änderung im Block vornehmen
30 BLWRITE: REM Veränderten Block an alte Position
schreiben
40 BLREAD 18,1,1024: REM Block 18,1 nach 1024
laden
50 BLREAD: REM Folgeblock nach 1024 laden
60 BLWRITE 3,4: REM Block aus 1024 nach 3,4
schreiben
```

TYPE (Gn) "Name" – gibt Floppydaten als ASCII-Zeichen aus

... liest die Datei »Name« aus und gibt die Daten-Bytes als ASCII-Zeichen auf dem Bildschirm aus.

Allgemeine Funktionen

... reichen von der Überprüfung auf angeschlossenen Geräte und deren letzte Funktion bis zur Zahlenformat-Umwandlung.

PRESENT (Gn) – Test auf angeschlossenes Gerät

... überprüft, ob das Gerät mit der Gerätenummer Gn angeschlossen ist. Beispiele: PRINT PRESENT (8) hat die Ausgabe von »1« zur Folge, wenn das entsprechende Gerät angeschlossen ist (sonst Ausgabe von »0«).

```
IF PRESENT (4)=0 THEN PRINT "Drucker nicht
eingeschaltet!"
```

EOF – End Of File

... ergibt eins, wenn das Dateiende erreicht ist, sonst ist die Ausgabe null. Beispiel:

```
10 OPEN 1,8,2,"test,s,r"
20 LABEL lesen: GET#1,A$
30 PRINT A$+CHR$(0)
```

```
40 IF EOF=0 THEN JUMP lesen
50 CLOSE 1
```

IORESULT (Gn) - letzte Anweisung (INPUT/OUTPUT)

... enthält das Ergebnis der letzten Eingabe/Ausgabe-Anweisung. Beispiele: PRINT IORESULT liest den Fehlerkanal der Floppy und gibt die Fehlernummer aus. Ist kein Fehler aufgetreten, so ist IORESULT Null.

```
10 BLOAD "Teil",49152
20 IF IORESULT=62 THEN PRINT "Datei nicht gefun-
den!":END
30 SYS 49152
```

Die Fehlernummern (beispielsweise 62 = File not found) sind dem Floppyhandbuch zu entnehmen.

BLOCKS (Gn) - Anzahl der freien Blocks

Gibt die Anzahl freier Blocks der Diskette aus. Beispiel: DNEW "Test,01":PRINT BLOCKS führt zur Ausgabe 664.

DSCPEEK (Gn) Adresse - PEEKen aus Floppyspeicher

... entspricht dem Basic-Befehl PEEK, nur wird hier auf den Floppyspeicher zugegriffen. Zu beachten: Die Adresse ist ohne Klammern zu schreiben. Beispiel:

```
A=DSCPEEK 49408:PRINT A
```

ergibt 120, da in der Adresse 49408 ($C100) der Wert 120 steht.

START (Gn) "Name" - Startadesse eines Programms

... holt die Startadresse eines Programms. Beispiel: PRINT START "Test" gibt die Startadresse des Programms »Test« aus.

! - dezimal nach hexadezimal

Umwandlung einer Dezimalzahl von 0 bis 65535 in eine zwei-Byte-Hexadezimalzahl und eine 16-Bit-Binärzahl. Beispiele:

```
PRINT !65535
$FFFF = %1111 1111 1111 1111
PRINT !0
$0000 = %0000 0000 0000 0000
```

Achtung: Das Ergebnis dieser Funktion ist ein String!

```
A$=!65534:PRINT A$
```

ergibt $FFFF = %1111 1111 1111 1111,

```
PRINT (LEFT$(A$,5)
```

ergibt $FFFF

$ - hexadezimal nach dezimal

... wandelt eine beliebige Hexadezimalzahl in eine Dezimalzahl um. Beispiel: PRINT $12345 ergibt 74565

% - binär nach dezimal

Umwandlung einer beliebigen Binärzahl in eine Dezimalzahl. Beispiel: PRINT %1111 ergibt 15

Achtung: Die Umwandlungsfunktionen können auch in Befehlen verwendet werden. Beispiel:

```
BLOAD "Test",$C000
DSCPOKE $77,$20+9
FOR x= $C000 TO $C100
POKE $D018,%10001111
etc.
```

Speicherbelegung von Disc-Basic:

Teil 1: $C000-$CFA2
Teil 2: $F72C-$FBA3
Funktionstastentexte: $EEBB-$EFBA
Diskettenblock : $EFBB-$F0BA

Fehlerbehandlung

ERROR-/+ - Fehlerbehandlung ja/nein

... schaltet auf eine programmgesteuerte Fehlerbehandlung um. Tritt ab dann ein Fehler im Programm auf, so wird die Routine angesprungen, die durch das LABEL error gekennzeichnet ist. Ist diese Routine nicht vorhanden, so wird im Programm fortgefahren. Die Fehlernummer bzw. die Zeile, in der der Fehler aufgetreten ist, sind in den Integervariablen ER% und EL% abgelegt. ACHTUNG: Tritt ein Fehler auf, so werden alle Rücksprungadressen (von GOSUB oder SUB) gelöscht. Mit ERROR+ wird auf die normale Fehlerbehandlung zurückgeschaltet. Bei Beendung des Programms geschieht dies automatisch.

Weitere Änderungen

LOAD und SAVE - neue Syntax

```
LOAD (Gn) "Test" entspricht LOAD "Test",Gn
LOAD "Test" entspricht LOAD "Test",8
SAVE (Gn) "Test" entspricht SAVE "Test",Gn
SAVE "Test" entspricht SAVE "Test",8
```

Ein Zugriff auf die Datasette ist NICHT mehr möglich, auch die RS232- (V24-) Routinen stehen nicht mehr zur Verfügung.

<SHIFT> + <RUN/STOP> - Directory

Diese Tastenkombination ist mit dem DIR-Befehl belegt.

<SHIFT> - Anhalten der LIST-Ausgabe

Bei LIST kann die Ausgabe durch Drücken der SHIFT-Taste angehalten werden. Bei Loslassen der Taste wird das LISTen fortgesetzt.

Neue Fehlermeldungen

BLOCK TOO LARGE (Nr. 30)

Der zu verschiebende Block bei GATHER überschreitet den Blockspeicher von 4 KByte.

FILE EXISTS (Nr. 31)

Der Name der relativen Datei, die angelegt werden soll (RELNEW), existiert schon auf der Diskette.

OVERFLOW IN RECORD (Nr. 32)

Die maximale Datensatzlänge wurde überschritten.

UNDEF'D RECORD (Nr. 33)

Die Recordvariable, die bei RELREAD/RELWRITE verwandt wurde, ist nicht definiert.

ILLEGAL FILE NAME (Nr. 34)

Bei Anlegen oder Öffnen einer relativen Datei wurden im Namen sogenannte 'Wild cards' (*, ?) verwendet.

FILE RESERVED (Nr. 35)

Die logische File-Nummer 1 ist für die relative Datei reserviert. Beispielsweise führt

```
10 RELOPEN "Test"
20 OPEN 1,8,2,"Name"
```

zu obengenannter Fehlermeldung, da die logische File-Nummer 1 schon von RELOPEN benötigt wird. Ist keine relative Datei geöffnet, kann die logische File-Nummer 1 verwendet werden.

Kassetten und RS232-Zugriff

OPEN 1,1,0 oder OPEN 1,2 führt zur Fehlermeldung ILLEGAL DEVICE NUMBER ERROR, da diese Routinen unter Disc-Basic nicht mehr zur Verfügung stehen.

Bereits der Anleitung können Sie entnehmen, welch ein umfangreiches Werkzeug Ihnen mit Disc-Basic gegeben wird. Mit Disc-Basic bekommt das Floppy-Handling eine völlig neue Bedeutung: Was bisher mühsam war, ist jetzt kinderleicht. (gr)

Kurzinfo: Disc-Basic

Programmart: Basic-Erweiterung
Laden: LOAD"DISC-BASIC",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 22
Programmautor: Gerald Kühne

Testbild V2.1 – TV am Monitor

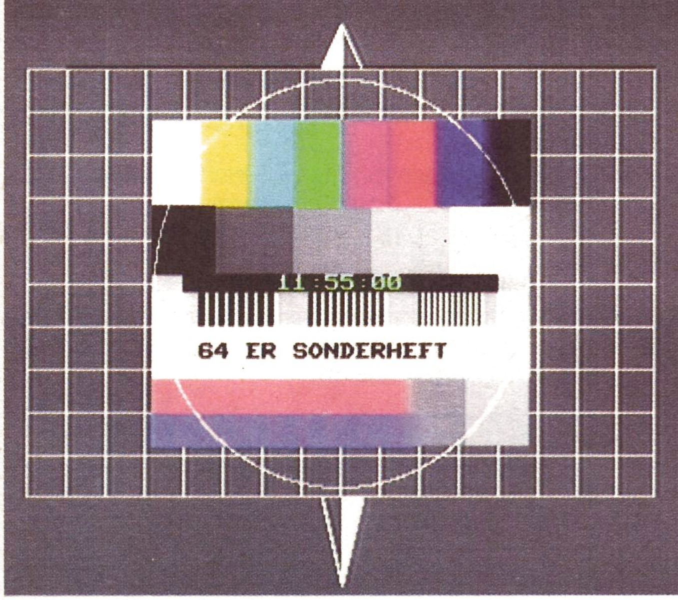
Testbilder aus dem Computer

Egal, wie Sie Ihren Fernseher auch einstellen: es kommt immer einer, dem ist das Bild zu blaß, zu bunt, zu unscharf, zu hell oder zu dunkel. Mit dem Programm »Testbild V2.1« können Sie Nörgler entwaffnen.

Dunkelblauer Hintergrund mit hellblauer Schrift und hellblauem Rahmen, so meldet sich der C64 nach dem Einschalten. Nicht gerade optimale Farben, um einen Farb- oder Schwarzweiß-Monitor richtig einzustellen. Spätestens nach dem Laden des Arbeitsprogramms wird also nachgeregelt und sollten sich hier die Farben ändern, geht dasselbe Spiel von vorne los.

Wir haben eine Ideallösung – den Testbildgenerator per Programm. Er erlaubt eine optimale Einstellung von Farbe und Kontrast. Aber nicht nur das, zusätzlich laufen Funktionen ab, die sonst nur in sündhaft teuren Meßgeräten eingebaut sind. »Testbild V2.1« nutzt alle Fähigkeiten des VIC. Leider, leider lassen sich aber einige hardwarespezifischen Eigenschaften nicht abstellen da der VIC nicht ganz nach der in Deutschland üblichen PAL-Norm arbeitet (s. Textkasten).

Trotz der Einschränkungen läßt sich der Testbildgenerator hervorragend zum Abgleich der Konvergenz (Farbreinheit) und Liniarität (Entzerrung) eines Fernsehers einsetzen (Abb.). Durch Zentrierungspfeile ist ebenfalls die Bildlage exakt justierbar und nicht zuletzt checken Pegeltöne den »Guten Ton«.



Das vom Fernsehen bekannte Testbild am C64

Unterschiede zwischen C-64-Videosignal und PAL-Norm

Bei der deutschen PAL-Norm (Puls-Alternating-Line, Zeilensprungverfahren) wird zwar ein Bild von oben links bis unten rechts aufgebaut. Es hat aber 625 Zeilen, von denen wechselweise 312 und 313 Zeilen übertragen werden. Der VIC dagegen sendet konstant 280 Zeilen zum Monitor.

Die Unterschiede zu PAL verkraften zwar die meisten Fernseher, aber Videorecorder der neuesten Generation sind überfordert. Sie sind auf ein normgerechtes Signal angewiesen.

Entsprechende Testbilder stehen Ihnen zur Verfügung. Geladen wird mit:

```
LOAD"TESTBILD V2.1",8
```

und gestartet mit RUN. Danach sehen Sie das Hauptmenü. Aus diesem lassen sich die einzelnen Funktionen über Tastendruck anwählen.

Farbabgleich

<Pfeillinks> – Weiß
... erzeugt einen weißen Bildschirm. Damit läßt sich ein Weißabgleich realisieren.

<1> – Rot
... schaltet den gesamten Bildschirm auf die Grundfarbe Rot. Diese Einstellung dient zur Prüfung der RGB-Endstufen und der Farbreinheit. Dasselbe gilt für
<2> – Grün und
<3> – Blau

Konvergenz und Linearität

<4> – Linien
... ergibt ein horizontales Linienmuster. Damit lassen sich horizontale Konvergenz und da die Linien im gleichen Abstand zueinander erscheinen, auch vertikale Linearität.

<5> – Kreis/Gitter
... erzeugt ein Gitter mit einem großen Kreis. Hiermit überprüfen Sie Konvergenz und Linearität.

<6> – Schachbrettmuster
... bringt wie der Name sagt ein gleichmäßiges Muster aus schwarzen und weißen Feldern. Damit läßt sich ebenfalls die Linearität einstellen.

Farbeinstellung

<7> – Farbbalken
... zeigt die berühmten Farbbalken, wie sie auch von Profi-

SONDERHEFT 78 VORSCHAU

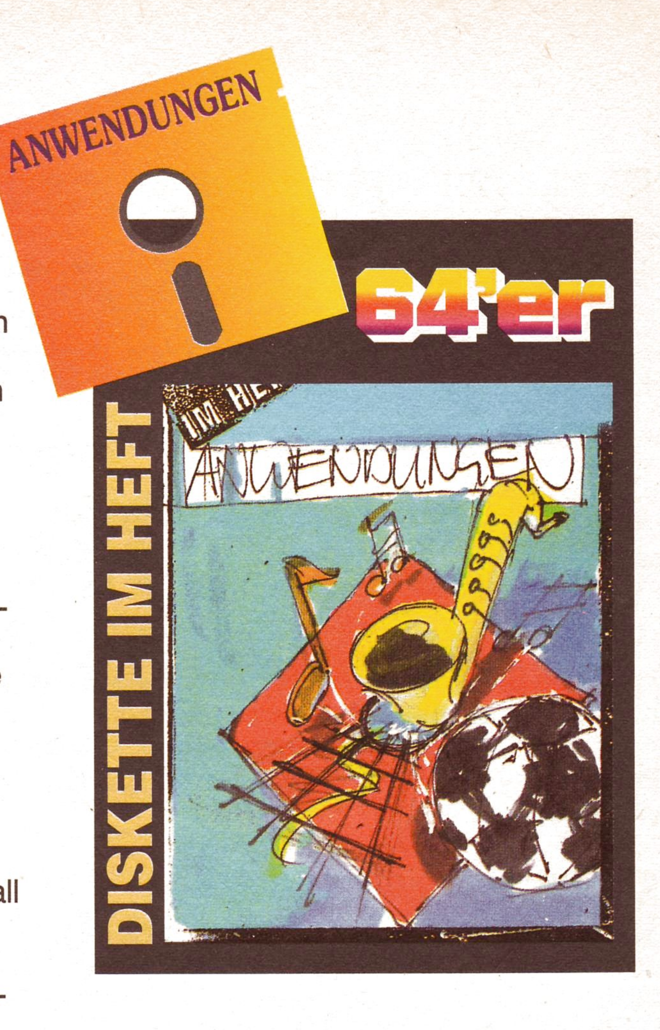
Nicht alltägliche, aber um so praktischere Anwendungen sind zentrales Thema unseres 64'er-Sonderhefts 78:

■ - Gratwanderung zwischen Einsen und Sechsen? »MAS 1.0« berechnet unbestechlich die Durchschnittsnoten in allen Fächern - auch, wenn's weh tut!

■ - Es muß nicht immer Hi-Eddi sein... »Mono Magic«, das Zeichenprogramm der dritten Art, entwirft nicht nur Supergrafiken auf dem Bildschirm, sondern staucht, streckt, verzerrt und wellt auch bereits fertige Hires-Bilder! Zaubern Sie die tollsten Grafikkreationen auf den Bildschirm!

■ - Dortmund vor, noch ein Tor! Am 15.8.92 beginnt die neue Fußballbundesliga-Saison, diesmal nur mit 18 Vereinen. Wer immer hautnah am Ball bleiben will, benutzt die komfortable Tabellenverwaltung »Bundesliga 2.0«! Alle Spieltage, Tabellen und Torschützen auf einen Blick!

■ - Einen Hauch von KI (Künstliche Intelligenz) bringt »Power of Logic« auf den Bildschirm: Je nachdem, wie informativ Ihre Eingaben waren, löst es die verzwicktesten Logic-Rätsel!



Ihr Sonderheft 78 liegt ab 22.5.92 bei Ihrem Zeitschriftenhändler.

Aus aktuellen oder technischen Gründen können einzelne Themen verschoben werden. Wir bitten um Verständnis.

kameras bekannt sind. Mit Ihrer Hilfe lassen sich Farbe und Farbkontrast abgleichen.

Technische Prüfungen

<8> – Multiburst

... erzeugt vertikale Linienmuster von 1 MHz bis 4 MHz. Dient zur Justierung der Bildschärfe und zur Beurteilung der max. Auflösung.

<9> – 2T-Impuls

... bringt eine feine vertikale Linie auf die Bildschirmmitte. Mit dieser Linie werden Laufzeiten und Reflexionen auf dem Übertragungsweg sichtbar.

<0> – 50-Hz-Sprung

... erzeugt ein weißes Fenster. Das Fenster muß in der Bildschirmmitte stehen. Bei defekten Klemmschaltungen im Synchrondetektor ist dies nicht der Fall.

<+> – Testbild

... zeigt ein Testbild für die allgemeine Qualitätsprüfung. In diesem Bild lassen sich auch mit <Z> die Uhrzeit und mit <T> die Textzeile ändern. Wenn Sie das Testbild erstmals anwählen, befindet sich das Programm automatisch in der Zeiteingabe.

Audio-Funktionen

<F1> – Pegelton

... am Lautsprecher muß ein 1000-Hz-Ton hörbar werden. Die Lautstärke läßt sich über die Tasten <CRSR rechts> zurücknehmen und mit <CRSR abwärts> erhöhen. Ein weiterer Tastendruck schaltet den Pegelton ab.

<F3> – Zweitonkennung

... erzeugt einen Ton mit der Frequenz von 274 Hz. Er schaltet einen Stereofernseher auf Zweikanal-Empfang um, bzw. ermöglicht den Abgleich dafür.

<F5> – Stereokennung

... bringt einen Ton mit der Frequenz von 117 Hz. Er schaltet einen Stereofernseher auf Stereo um, bzw. ermöglicht den Abgleich dafür.

Sonderfunktionen

<SPACE> – Menü

... ruft aus jeder Funktion ein Übersichtsmenü auf.

<CTRL/RESTORE> – Programmende

... beendet »Testbild V2.1«. Neustart ist mit »SYS 13824« möglich.

Mit den Funktionen des Testbildgenerators läßt sich jeder Fernseher oder Monitor auf optimale Wiedergabequalität justieren. Aber auch bei der Fehlersuche erweist sich dieses Programm als überaus nützlich. (gr)

Kurzinfo: Testbild V2.1

Programmart: Testbildgenerator
Laden: LOAD"TESTBILD V2.1",8
Starten: nach dem Laden RUN eingeben
Benötigte Blocks: 29
Programmautor: Mario und Maik Qualmann

Der erste Spiele-Hit des Jahres vom 64'er Team.



Weitere Highlights:

- »Marktübersicht« – die aktuellsten 64er Computer-Spiele im Überblick.
- »Longplays« – "Zac McKracken" und "Bard's Tale" werden Schritt für Schritt durchgespielt.
- »Tips & Tools« – 10 Trainerprogramme und jede Menge Cheats machen fit für den Top-Level.

Jetzt bei Ihrem Zeitschriftenhändler!